DIÊGO FIALHO RODRIGUES

ACOMPANHAMENTO DO DESENVOLVIMENTO PROFISSIONAL DE EGRESSOS POR MEIO DE SISTEMAS MULTIAGENTES

VIÇOSA
MINAS GERAIS - BRASIL
2014

DIÊGO FIALHO RODRIGUES

ACOMPANHAMENTO DO DESENVOLVIMENTO PROFISSIONAL DE EGRESSOS POR MEIO DE SISTEMAS MULTIAGENTES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2014

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca Central da UFV

T
R696a
2014

Rodrigues, Diêgo Fialho, 1984-
Acompanhamento do desenvolvimento profissional de egressos por meio de sistemas multiagentes / Diêgo Fialho Rodrigues. – Viçosa, MG, 2014.
xiii, 88f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.
Orientador: Alcione de Paiva Oliveira.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f.80-85.

1. Inteligência artificial. 2. Processamento de linguagem natural (Computação). 3. Sistemas multiagentes. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-graduação em Ciência da Computação. II. Título.

CDD 22. ed. 006.3

DIÊGO FIALHO RODRIGUES

ACOMPANHAMENTO DO DESENVOLVIMENTO PROFISSIONAL DE EGRESSOS POR MEIO DE SISTEMAS MULTIAGENTES

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

APROVADA: 13 de fevereiro de 2014.

Levi Henrique Santana de Lélis

Regina Maria Maciel Braga Villela

Alcione de Paiva Oliveira
(Orientador)

Dedico esta dissertação ao meus pais Milton e Dodora
pelas oportunidades e incentivos para estudar que nunca me faltaram.

Agradecimentos

Agradeço primeiramente ao meu Orientador, Alcione, por ter me guiado durante todo o percurso. Também agradeço a compreensão em relação a escassez de tempo pelo fato de eu ter que conciliar trabalho e estudo. Por falar em trabalho, gostaria de agradecer ao pessoal da DTI que nunca imposeram barreiras para que eu pudesse concluir meu mestrado. Acredito que o que aprendi neste período vai trazer grandes benefícios ao meu desempenho profissional. Pela paciênica, agradeço à minha namorada que acabou virando esposa neste período. Foram muitos sábados, domingos, feriados e noites dedicados ao estudo e que não pude dar toda a atenção que você merecia, Fátima.

Todas as pessoas com quem convivemos acabam influenciando na sua vida, positiva ou negativamente. Sempre adimirei pessoas que lutam com muita dificuldade para conseguir vencer na vida e que subvertem todas as estatísticas e leis da probabilidade. Neste quesito, considero-me iluminado. Sempre tive pais que me apoioaram nos estudos, desde o meu primeiro dia de aula, no extinto pré de seis. O seu Milton e a dona Dodora sempre me deram todas as condições para estudar e isso eu vou valorizar a minha vida inteira.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

RODRIGUES, Diêgo Fialho, M.Sc., Universidade Federal de Viçosa, Fevereiro de 2014. **Acompanhamento do Desenvolvimento Profissional de Egressos por meio de Sistemas Multiagentes.** Orientador: Alcione de Paiva Oliveira.

Várias instituições precisam coletar e armazenar informações sobre pessoas relacionadas com suas atividades. Dentre essas instituições destacam-se as instituições de ensino, que necessitam acompanhar a evolução profissional de seus egressos. Existe uma vasta quantidade de informação sobre indivíduos disponíveis publicamente em diversos formatos e acessadas de uma maneira não estruturada: redes sociais, revistas eletrônicas, bases de currículos (e.g. Lattes), sites de busca, etc. Devido a esta diversidade de fontes, formatos e dimensão é impossível manter manualmente uma base de dados com informações atualizadas. Por isso, um sistema que automatizasse a extração e armazenamento de tais informações poderia tornar este trabalho menos maçante. Contudo, um sistema automatizado para realizar tal tarefa precisa superar vários desafios: coletar informações em bases de textos em linguagem natural, efetuar a extração de informação por meio de técnicas de processamento de linguagem natural, identificar entidades e os papéis exercidos por essas entidades na situação descrita, integrar informações heterogêneas e armazenar essas informações em uma base de fácil manipulação. A tecnologia de Sistemas multiagentes (SMA) se apresenta como uma alternativa para resolver este tipo de problema de uma maneira gradual e escalável. SMA permitem a construção de sistemas que precisem atuar de forma distribuída e descentralizada e que permita o acréscimo de elementos de processamento de forma transparente e incremental. Este trabalho explora o uso de SMA para fazer o acompanhamento da trajetória profissional de egressos utilizando técnicas de coleta de informações e processamento de linguagem natural com frames semânticos e ontologias.

Abstract

RODRIGUES, Diêgo Fialho, M.Sc., Universidade Federal de Viçosa, February, 2014. **Follow-up of Graduates with Multiagent Systems.** Adviser: Alcione de Paiva Oliveira.

Several institutions need to collect and store information about people related to their activities. Among these institutions there are the educational institutions that need to monitor the professional development of its graduates. There is a vast amount of information about individuals publicly available in several formats and accessed in a non-structured way: social networks, electronic journals, curricula databases (eg Lattes), search engines, etc. Due to this diversity of sources, formats, and size it is impossible to manually maintain a database with updated information. Therefore, a system that automates the extraction and storage of such information could make this work less dull. However, an automated system to perform this task must overcome several challenges: collect information in databases in natural language texts, perform the information extraction through natural language processing techniques, identify entities and the roles played by these entities in the situation described, integrate heterogeneous information and store that information in a database of easy handling. The technology of Multiagent systems (MAS) is presented as an alternative to solve this problem in a gradual and scalable way. MAS allow the construction of systems that need to operate in a distributed and decentralized fashion and allowing the insertion of processing elements transparently and incrementally. This work explores the use of SMA to track the career paths of graduates using techniques of information gathering and processing of natural language with semantic frames and ontologies.

Capítulo 1

Introdução

O acompanhamento de pessoas é uma tarefa de grande valor para as instituições. Cada indivíduo tem suas próprias preferências, gostos e características. As pessoas também possuem uma trajetória de vida e um histórico. Saber extrair e utilizar estas informações pode trazer diversos benefícios para quem as analisa. São diversos os exemplos mostrando a importância de se acompanhar pessoas. Uma empresa pode estar interessada em captar as preferências de seus clientes, ou então expandir seu mercado para novas tendências. Ou então uma companhia pode buscar informações sobre pessoas de acordo com as habilidades e formação para preencher uma vaga de trabalho.

Particulamente, o acompanhamento de egressos é de grande importância para as instituições de ensino. O sucesso profissional das pessoas formadas por estes estabelecimentos pode ser atribuído, em parte, à formação recebida nessas instituições. Um dos critérios estabelecidos pelo Ministério da Educação para avaliar as instituições de ensino é o acompanhamento de seus egressos. No entanto, essa é uma tarefa quase impossível de ser realizada em sua totalidade, e mesmo em uma percentagem significativa. É preciso enviar correspondência de tempos em tempos para endereços possivelmente desatualizados, solicitando que os ex-alunos atualizem seus dados e endereços: é uma contradição. Desta forma, seria interessante usar um sistema que fosse capaz de automatizar parte deste processo e que fosse capaz buscar informações no maior número de

fontes possível. Este trabalho apresenta um modelo de sistema multiagente [Wooldridge, 2002] que trabalha utilizando-se de várias técnicas, desde busca e gerenciamento de informações até processamento de linguagem natural.

1.1 O Problema e sua Importância

Com o advento da Internet, o mundo nunca esteve tão repleto de informações sobre pessoas. Os usuários estão conectadas boa parte do tempo, disponibilizando informações em redes sociais, listas de discussão, blogs, revistas eletrônicas, bases de currículos, etc. Paradoxalmente, buscar estes dados tem se tornado cada vez mais difícil. São diversas fontes espalhadas por toda a Web. Buscar estas informações de forma manual se torna uma tarefa maçante e praticamente impossível de ser feita em sua completude. Sendo assim, a automatização deste tipo de trabalho através de computadores se torna uma alternativa interessante.

Se por um lado esta tarefa é inviável para ser feita de forma manual, é também um desafio para sistemas de informação. São vários obstáculos a serem vencidos. Como dito anteriormente, são diferentes fontes e cada uma com seu formato próprio. Na maioria das vezes, a informação se encontra em formato não estruturado, geralmente em linguagem natural. Além disso, computadores não possuem o conhecimento contextual que as pessoas possuem, impossibilitando a realização de inferências de senso comum.

Para auxiliar na resolução deste problema, os Sistemas Multiagentes [Wooldridge, 2002; Bordini et al., 2006, 2007] aparecem como uma alternativa para abordar esse problema de uma maneira gradual e escalável. Sistemas multiagentes permitem a construção de sistemas que precisem atuar de forma distribuída e descentralizada e que permitam o acréscimo de elementos de processamento de forma transparente e incremental.

O presente trabalho foca particularmente no acompanhamento da evolução profissional de egressos. Todo o desenvolvimento é feito sobre uma plataforma multiagente que possibilita a agregação de diferentes soluções, como processamento de linguagem

natural, gerência de bases dados e coleta de informações na Internet.

1.2 Hipótese

É possível acompanhar e armazenar informações a respeito da trajetória profissional de egressos através do uso de Sistemas Multiagentes com extração de dados na Internet e utilização de técnicas de processamento de linguagens naturais.

1.3 Objetivos

O principal objetivo deste trabalho é desenvolver um modelo de Sistema Multiagente que seja capaz de realizar o acompanhamento de egressos. O sistema deve buscar evidências de ligação entre empresas e ex-alunos e armazenar as informações de forma estruturada.

Além do objetivo principal, temos os seguintes objetivos secundários:

- Extender uma ontologia [Chandrasekaran et al., 1999] capaz de descrever o domínio do problema em questão, como os conceitos de empresas, cargos, pessoas, lugares, etc; assim como seus relacionamentos;

- Identificar padrões de troca de mensagens entre egressos que evidenciem a ligação empregatícia destes e, à partir disto, desenvolver frames semânticos [Baker et al., 1998] descrevendo tal cenário;

- Desenvolver um modelo de redes bayesianas [Friedman et al., 1997] para cada frame semântico criado afim de fazer a avaliação das mensagens.

Para fornecer evidências de modo a corroborar a hipótese, foi realizado um estudo de caso envolvendo o acompanhamento da trajetória profissional dos egressos do curso de ciência da computação da Universidade Federal de Viçosa.

1.4 Organização da dissertação

Esta dissertação esta organizada sob a forma de coletânea de artigos. Este formato está de acordo com as normas da Comissão do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Viçosa. Ao todo são três artigos. O primeiro artigo, intitulado *Semi-automatic Follow-up of Graduates*, foi publicado nos anais da XXXI *International Conference of the Chilean Computer Science Society* (SCCC 2012). O segundo artigo, intitulado *Applying ontology to align natural language sentences and frames*, ainda não foi submetido. O terceiro artigo, intitulado *Acompanhamento da Evolução Profissional de Egressos com Sistemas Multiagentes*, foi enviado para a avaliação da Revista para o Processamento Automático das Línguas Ibéricas (Linguamática).

A dissertação esta organizada da seguinte forma: o capítulo 1 faz uma introdução ao problema e descreve sua importância. Este capítulo também apresenta a hipótese e objetivos da pesquisa. O segundo capítulo apresenta o referencial teórico, dando uma breve explicação para os tópicos abordados neste trabalho. O terceiro capítulo transcreve os três artigos, resultado desta pesquisa de mestrado. O Artigo I traz uma proposta inicial para o problema de acompanhamento de egressos com elucidação do modelo e exemplos. O Artigo II foca no processamento de linguagem natural com Frames Semânticos e ontologias. O Artigo III faz uma apanhado geral do trabalho utilizando todas as soluções pesquisadas e apresenta testes e resultados do modelo proposto. No Capítulo 4 são apresentadas as conclusões gerais, principais dificuldades e propostas para trabalhos futuros.

Capítulo 2

Referencial Teórico

Este capítulo tem a intenção de servir como referencial para o capítulo 3 que apresenta os artigos produzidos. Basicamente, são abordados 4 temas que servem para o melhor entendimento do assunto. O primeiro tema são os Sistemas Multiagentes, abordagem que foi utilzada em nosso modelo. O segundo tema, Ontologias, foi utilizado para formalizar todos os conceitos e relações existentes no domínio da aplicação. Na parte referente à linguística e de processamento de linguagem natural, o Frames Semânticos foram utilizados para definir padrões de enunciados e este tema também é abordado. O último tema deste capítulo são as Redes Bayesianas, utilizadas para fazer uma avaliação probabilística das sentenças que foram processadas.

2.1 Sistemas Multiagentes

Sistema Multiagentes (SMA) é uma área dentro da computação que trata de aspectos de computação distribuída aplicados à sistemas de inteligência artificial [Wooldridge, 2002]. Basicamente, um SMA é formado por agentes, que são entidades de software autônomas, e o ambiente, local onde os agentes atuam. Os agentes dentro de um ambiente formam uma sociedade. Eles podem interagir entre si, alterando o estado do ambiente, afim de alcançar seus objetivos.

De acordo com Russell et al. [1995], os ambientes podem ser classicados segundo as

seguintes propriedades:

- *Acessível x Inacessível:* um ambiente acessível é aquele onde o agente pode obter todas as suas informações. Nos ambientes inacessíveis, essa informação é limitada.
- *Determiniístico x Não-determinístico:* nos ambientes determinísticos se tem uma noção exata dos efeitos de cada ação.
- *Estático x Dinâmico:* os ambientes estáticos não sofrem alterações em seus estados a não ser por ações tomadas por agentes. Em um ambiente dinâmico, existem outros processos alterando o ambiente que fica além do controle dos agentes.
- *Discreto x Contínuo:* um ambiente discreto possui um número fixo e finito de ações e percepções.

Em se tratando de agentes, existem dois grandes tipos: os reativos e os cognitivos [Hübner et al., 2004]. O primeiro tipo não faz nenhuma deliberação em relação ao estado do ambiente onde atua. Os agentes reativos simplesmente percebem o ambiente e, de acordo com o estado atual deste ambiente, tomam uma atitude agindo de forma totalmente reativa. Em SMA reativos, o comportamento inteligente é atingido da interação de um grande número de agentes muito simples.

Sobre agentes cognitivos, a definição não é consenso total entre os autores. Porém, alguns aspectos são considerados importantes. Desta forma, agentes cognitivos seriam entidades capazes de: perceber o ambiente, alterar o ambiente, comunicar-se com outros agentes, representar internamente o ambiente percebido, possuir desejos ou objetivos, utilizar-se de técnicas de raciocínio e aprendizagem, tomar decisões de acordo com seus objetivos e possíveis cenários no futuro (deliberação).

Três atitudes mentais são especialmente importantes na construção de arquiteturas de agentes deliberativos: as crenças (beliefs), os desejos (desires), e as intenções (intentions) [Guerra-Hernández et al., 2005]. Destes conceitos surgem as arquiteturas BDI (beliefs, desires, intentions). As crenças representam o conhecimento que o agente tem sobre o estado do ambiente e de outros agentes. As crenças são formadas através da

percepção do ambiente e da consulta de crenças já existentes. Os desejos são os estados do ambiente que o agente quer atingir. Os objetivos são um subconjunto dos desejos que não são mutuamente exclusivos. As intenções representam uma sequências de ações necessárias para se atingir certo objetivo.

2.1.1 A Comunicação entre Agentes

A comunicação tem um papel central em SMA. Agentes realizam ações coordenadas, seja por meio da cooperação, seja por meio da competição (negociação). A comunicação é o meio pelo qual os agentes interagem em uma sociedade e buscam atingir seus objetivos. Os estudos da área de comunicação são embasados na teoria dos atos de fala [Austin, 1975]. Esta teoria define a linguagem humana como ação. Em contraposição à trabalhos de semântica lógica, a teoria dos atos de fala afirma que algumas sentenças não podem ser analisadas em termos de condições de verdade (sentenças performativas). Ainda de acordo com a teoria, uma sentença é composta do conteúdo semântico e da intenção do falante. Desta forma, um mesmo conteúdo semântico pode estar presente em diferentes situações comunicativas, diferindo apenas pelo que se chama força ilocucionária (intenção do falante). A força ilocucionária pode ser classificada em: assertivas ou representativas, diretivas, comissivas, expressivas e declarativas.

A classificação de mensagens na teoria dos atos de fala depende de vários fatores como papéis sociais, normas e convenções. Posto isto, a semântica da comunicação deve ser analisada dentro do contexto de um diálogo, e não de forma isolada. Em SMA, protocolos são usados para padronizar sequências típicas de troca de mensagens.

A Linguagem de Comunicação de Agentes (ACL) é uma tecnologia para comunicação entre agentes [Chaib-draa and Dignum, 2002]. Ela foi a primeira a incluir conceitos complexos de comunicação em alto nível. A ACL é composta de três partes: sistema ontolingua, que trata de ontologia de vocabulários; Knowledge Interchange Format (KIF) [Genesereth et al., 1992], uma extensão de uma lógicas de primeira ordem para codificar as mensagens; e a Knowledge Query and Manipulation Language (KQML) [Finin et al., 1994], uma linguagem para inserir contexto às mensagens.

KQML é baseada na teoria dos atos de fala. A linguagem possui um conjunto de performativas que tem a função de explicitar a intenção do agente remetente da mensagem. A KQML utiliza-se também de um conjunto especial de agentes, chamados de facilitadores. Estes agentes tem a função de mediadores, tendo conhecimento dos agentes acessíveis e das habilidades de cada um. Uma mensagem KQML é formada de: sender, receiver, language, ontology, content. As mensagens, neste caso performativas, são divididas em várias categorias. Dentre elas, as principais são: tell, deny e untell (de informação); ask-if e ask-all (de consulta); error e sorry (respostas básicas); insert e delete (de base de dados); advertise (definição de capacidades); achieve e unachieve (de efetuação); register, unregister, transport-address e broadcast (de rede); e finalmente broker-one e recommend-one (de facilitação). As mensagens são trocadas respeitando um protocolo, ou seja, existe uma sequência determinada de troca de mensagens.

É necessário que todos os agentes de um determinado ambiente estejam de acordo com o significado das mensagens trocadas. As ontologias são usadas neste sentido. Elas fornecem um vocabulário comum para representar o conhecimento do domínio. As ontologias são essenciais para o desenvolvimento de sistemas inteligentes e também para comunicação entre sistemas heterogêneos.

As trocas de mensagens influenciam os conhecimentos e intenções dos agentes. Desta forma, faz-se necessário a utilização de uma semântica para a linguagem KQML para descrever os estados dos agentes antes e depois da comunicação. Esta descrição é dada em termos de pré-condições e pós-condições. As pré-condições indicam as condições necessárias para envio e recepção de performativas. As pós-condições descrevem os estados dos emissores e receptores após o envio da mensagem.

2.1.2 Engenharia de Software Orientada à Agentes

Sistemas Multiagentes se tornaram uma poderosa ferramenta para resolução de problemas de TI, que estão cada vez mais complexos. Desta forma, SMA tem se tornado um novo paradigma para desenvolvimento de software. Desenvolvimento de software orientado à agentes oferece autonomia aos componentes do sistema, flexibilidade, além

de uma forma de aplicar inteligência artificial à partes especificas do software. No entanto, ainda é um desafio unir os conceitos de SMA e Engenharia de Software, algo que tem sido alvo de várias pesquisas. Zambonelli and Omicini [2004] analisam tais desafios de acordo com a "escala de observação" que é usada para construir o sistema: micro, macro e meso.

A Engenharia de Software Orientada à Agentes (ESOA) [Bernon et al., 2005] é centrada no conceito do agente. Juntamente com a definição de agente aparecem novas abstrações que são importantes, tais como: autonomia, localização e socialização. Outro aspecto importante é que agentes não devem ser modelados apenas através de suas interações. Uma modelagem mais precisa deve focar também nos conceitos de ambiente e sociedade em um mundo em que agentes tem acesso a outros tipos de recursos e desempenham diferentes papéis. A partir destas considerações, a ESOA surge como um novo paradigma para desenvolvimento de SMA.

Em 30 anos de pesquisa em IA, os estudos estavam concentrados no aspecto científico da IA (visão artificial, representação do conhecimento e planejamento) e muito pouco no aspecto construtivo (engenharia de software). É neste ponto que agentes se tornam importantes na junção destes dois aspectos. O encapsulamento conseguido à partir do uso de agentes é uma forma de separar sistemas inteligentes em módulos e fazer com que estes sejam mais facilmente integrados. Desta forma, o uso de ESOA pode aumentar substancialmente o nível de complexidade que podem ser resolvidos através de SMA.

Mesmo que seja possível construir sistemas distribuídos complexos em termos de objetos e interações entre cliente e servidor, esta escolha não parece viável quando falamos de um SMA. Um novo conjunto de ferramentas conceituais e praticas é necessário para facilitar a construção de SMA. Com este intuito, muito se tem pesquisado e em especial nas seguintes linhas: modelagem de agentes, arquiteturas de SMA, metodologias para SMA, técnicas de notação e infraestrutura de SMA. Mesmo que estes trabalhos tenham contribuído para o estabelecimento da ESOA, os desafios nesta áreas são grandes e as linhas de pesquisa podem seguir várias outras direções além das citadas.

Uma questão que surge quando pensamos nas direções que os SMA devem tomar

é: o que significa aplicar engenharia a um sistema em cenários modernos do futuro? A mesma questão aparece na discussão sobre as direções que a pesquisa em ESOA deve tomar. Dispositivos computacionais estão cada vez mais presentes no nosso cotidiano. Todos estes componentes interagem em num contexto dinâmico e complexo. À partir desta perspectiva, o conceito de sistema de torna obscuro uma vez que cada componente pode ser desenvolvido ou instalado a partir de componentes já existentes. Assim sendo, fica difícil definir os limites dos componentes e em que escala estes serão aplicados. Então a questão levantada no início do paragrafo pode ser substituída por: em qual escala de observação que o trabalho de engenharia deveria se situar? Basicamente, podemos dividir as escalar em três: micro, macro e meso. Estas escalar serão abordadas nos próximos parágrafos.

A escala micro é assunto da vasta maioria das pesquisas na área de ESOA. De fato, até poucos anos atrás, sistemas de software trabalhavam de forma isolada ou se tinham alguma comunicação, era bem limitada. No entanto, para convencer aqueles que já usam métodos tradicionais de desenvolvimento (como orientação a objetos), é preciso apresentar evidências que o uso de ESOA pode ajudar a economizar dinheiro e recursos. O objetivo não seria mostrar as vantagens que agentes trariam para sistemas de software, mas os benefícios que estes trariam para o processo de desenvolvimento de software. Muita pesquisa tem sido realizada nesta área, mas ainda é preciso muito trabalho que mostrem resultados quantitativos mostrando as vantagens do paradigma ESOA.

A maioria das metodologias que descrevem como um SMA deveria ser construído é baseada no modelo cascata. Este fato levanta duas questões. Primeiro, os modelos tradicionais de desenvolvimento de software tem que se aplicar à SMA também? Segundo, ESOA não poderia se aplicar aos modelos de desenvolvimento ágil? Infelizmente, ainda não se tem uma resposta clara a estas duas perguntas indicando uma direção que as pesquisas poderiam tomar.

AUML [Odell et al., 2000] é a principal linguagem para modelagem de SMA baseada na UML. Apesar da AMUL exercer um papel importante na consolidação da ESOA,

a natureza complexa e dinâmica dos sistemas modernos não pode ser expressada por linguagens que foram projetadas para arquiteturas estáticas e não situadas. Desta forma, juntamente com a AUML, novas propostas seriam muito bem-vindas e não deveriam ser descartadas apenas pelo fato de não respeitarem certos padrões.

Métodos formais sempre tiveram um importante papel na engenharia de software. Com a complexidade dos sistemas modernos, ESOA representam uma nova oportunidade para estudos aplicados à escala micro. Neste nível, a complexidade é alcançada não apenas pela inserção de novos componentes, mas pelo aumento das capacidades dos componentes individualmente. Devido ao encapsulamento natural dos SMA, arquiteturas formais de agentes (como BDI e agentes lógicos) poderiam ser usados para construir componentes autônomos e complexos cujo comportamento poderia ser modelado através de técnicas de inteligência artificial tradicionais.

Um dos desafios ta ESOA hoje seria incorporar, de uma forma fácil, a engenharia de comportamentos inteligentes à engenharia de software tradicional. O objeto seria propiciar o uso de um arcabouço conceitual para técnicas de IA. Desta forma, engenheiros de software poderiam selecionar soluções de IA e incorporar aos seus sistemas de forma prática.

ESOA aplica à escala macro abstrai do fato de que um sistema global é composto de vários sub-sistemas e agentes. O que é relevante nesta escala é o comportamento do sistema como um todo. Muitos leitores alegam que a escala macro não tem ligação nenhuma com a engenharia, mas sim com a investigação científica. No entanto, sistemas de softwares chegaram a um nível de complexidade que é preciso aplicar alguma engenharia através de metodologias rigorosas para que possamos ter algum controle sobre estes artefatos.

A engenharia sempre esteve relacionada com o ato de medir. Geralmente, na escala macro, perdemos visibilidade e controle sobre componentes individuais. A única forma de caracterizar o sistema seria através do uso de métricas para capturar aspectos relevantes e comparar quantitativamente dois sistemas. Com a medição, poderíamos definir intervalos aceitáveis para o sistema e controlá-los para terem os comportamentos

esperados.

Outro aspecto da escala macro é a correlação com áreas como sociologia e biologia, e também com a universalidade que se pode aplicar à SMA. Muitos sistemas físicos exibem os mesmos comportamentos universais, que são tipicamente partes fracamente acopladas interagindo segundo alguma lei. O mesmo pode ser aplicado à SMA para construir ferramentas de engenharia de propósito geral. Outra questão é a aplicação que SMA podem fazer em áreas como sociologia e biologia. Isto se dá pelo fato de SMA serem naturalmente abstraídos em termos ecológicos e sociais.

A escala meso entra em cena quando as características de escala micro de um sistema tem que estar em harmonia com as características macro. Muitos trabalhos na área de ESOA tem tratado a escala mesmo de forma muito restrita, como se fosse apenas uma extensão da escala micro. Basicamente, dois aspectos que ainda precisam ser superados, que seria entender e controlar os impactos que a implantação de um sistema teria em uma escala macro, e entender e controlar os impactos que um sistema de software teria quando inserido num cenário de escala macro e vice-versa.

Em um cenário onde agentes ganham vida e morrem a qualquer hora, passam para diferentes domínios, e delegam tarefas para outro subsistemas, definir os limites de sistema de software se torna um problema. Apesar de algumas soluções terem sido apresentadas, uma direção deve ser tomada dentro da ESOA para se criar metodologias para identificação e controle dos limites de um SMA. Contudo, este problema nos levaria ao problema de formalização na escala meso.

Na escala meso os engenheiros de software não se podem dar ao luxo de perder controle sobre os componentes do sistema, independente da intrínseca explosão de complexidade. Algumas questões não podem ser resolvidas sem a ajuda de métodos tradicionais. No entanto, ao menos algumas partes do sistema podem ser modeladas formalmente. A infraestrutura de interação compartilhada é um exemplo. Se o comportamento dinâmico dos SMA puderem ser modeladas e previsíveis, o sistema global correspondente poderá ser projetado ao menos parcialmente independente do comportamento autônomo dos agentes.

Conceber modelos confiáveis para a engenharia de sistemas complexos é de suma importância para o progresso tecnológico dos SMA. Confiabilidade se tornou um dos mais importantes quesitos "sociais" em SMA e engloba duas questões: confiabilidade entre humanos e sistemas e confiabilidade entre sistemas. Com a interpretação de SMA em termos de sociedades, é possível encarar estas duas questões com o mesmo arcabouço conceitual, adotando uma abordagem uniforme para explorar modelos e soluções de forma geral.

Um desafio chave na área de ESOA é prover modelos, tecnologias e metodologias para prover inteligência social. Na escala meso, a inteligência embutida nos agentes geralmente não é suficiente para construir sistemas inteligentes. Desta forma, o desenho de sistemas inteligentes requer abstrações que suportem inteligência social e de uma infraestrutura que suporte ambientes onde agentes possam exercer tanto sua inteligência individual como sua inteligência social.

2.2 Ontologias

Pessoas, organizações e sistemas de TI precisam se comunicar de forma efetiva. Porém, em um ato de comunicação, um mesmo conceito pode ser referenciado de diferentes maneiras, nem sempre de conhecimento de todas as partes envolvidas na comunicação. Além disso, pode haver formas parecidas ou iguais de referenciar um mesmo conceito, necessitando de uma clareza maior sobre o contexto. Como proposto por Uschold and Gruninger [1996], uma forma de resolver tal problema seria reduzir ou eliminar estas confusões de conceitos e terminologias para se chegar a um entendimento comum, um arcabouço unificado de conceitos: uma ontologia.

Em termos filosóficos, ontologia é o estudo dos tipos de coisas que existem. Uma ontologia é uma visão de mundo de acordo com um determinado contexto. Esta visão de mundo é concebida atavés de conceitos, a definição destes conceitos e suas relações. Estas conceitualizações podem ser implícitas, existindo na mente de alguém ou pode ser representada através de um software.

Dentro da área de Inteligência Artificial, o termo ontologia é utilizado para designar um vocabulário de termos em um certo domínio, bem como suas relações. Uma ontologia também pode possuir um conjunto de termos que representam fatos sobre o domínio representado [Chandrasekaran et al., 1999]. De acordo com Uschold and Gruninger [1996], uma ontologia possui vários níveis de formalidade, dependendo da maneira que seus conceitos são especificados. Podem variar de representações totalmente informais (como a estruturada em formato de linguagem natural) até representações rigorosamente formais (com o uso de teoremas e semântica formal).

Ontologias ajudam a esclarecer a estrutura do conhecimento oferecento um volcabulário para representar o conhecimento [Chandrasekaran et al., 1999]. Por exemplo, podemos utilizar uma ontologia para descrever os conceitos existentes dentro de uma empresa. Uma companhia possui várias instalações, subclassificadas em matriz e filial. Cada uma destas instalaçãoes possui relacionamentos com localizações físicas, designando seu endereço. Uma empresa também possui relações com pessoas, como funcionários, donos, clientes. Os funcionários podem exercer vários papeis dentro desta empresa. Sem todos estes termos e relações corretamente definidos as chances de se ter uma base de conhecimentos errônea seria muito maior, levando a problemas de inconsistência.

O conhecimento também pode ser compartilhado através do uso de ontologias. Associando termos com conceitos e relações podemos codificar todo o conhecimento de um domínio em uma ontologia. Caso a ontologia tenha o objetivo de representar o conhecimento em um nível mais abstrato e genérico, podem ser usadas ontologias de topo. Ontologias de topo trazem definições de conceitos em um nível mais alto, provendo uma hierarquia básica de conceito e relações. Ontologias mais específicas podem ser acopladas à alguma ontologia de topo, aproveitando-se da estrutura já existente. Muitas ontologias possuem o conceito thing (coisa) ou entity (entidade) como classe raíz, mas podem divergir nos níveis seguintes. A figura 2.1 mostra alguns exemplos de ontologia do tipo topo, como CYC [Reed et al., 2002] e GUM (Generalized Upper Model) [Bateman et al., 1995].

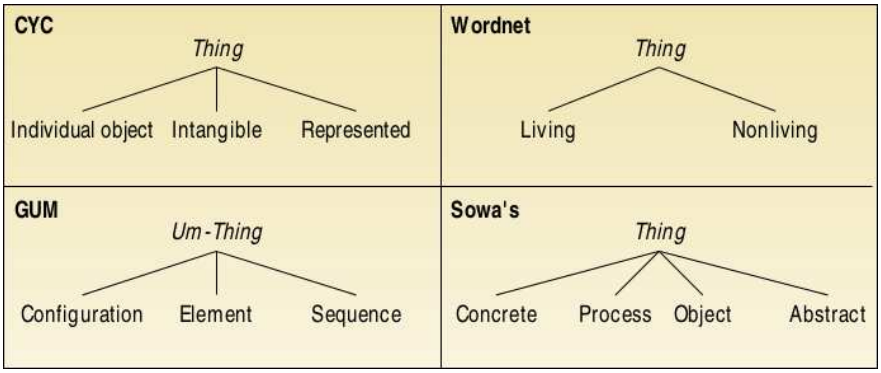

Figura 2.1: Alguns exemplos de diferença na hierarquia de ontologias de topo. Fonte: [Chandrasekaran et al., 1999].

2.3 Semântica de Frames

A Semântica de Frames é um estudo empírico em semântica que enfatiza a ligação entre a linguagem e seu significado [Petruck, 1996]. Neste contexto, um Frame pode ser entendido como um sistema de conceitos relacionados de forma que pra que se entenda um determinando conceito é necessário entender o sistema como um todo. Conceitos similares à Semântica de Frames foram desenvolvidos tanto dentro da área de inteligência artificial [Minsky, 1974] como em outras áreas, caso da psicologia cognitiva [Schank and Abelson, 1977].

Outra definição é apresentada por Baker et al. [1998]: Frames Semânticos são estruturas conceituais que descrevem um tipo particular de situação, objeto ou evento juntamente com seus participantes. A ideia pode ser exemplificada através do Frame Commerce_buy. Como o nome diz, este frame evoca um cenário de fundo de uma transação comercial, envolvendo elementos como Buyer (comprador), Goods (Bens), Money (dinheiro) e Seller (vendedor). Na sentença abaixo podemos ver como os constituintes do Frame são marcados:

[Most of my audio equipment $_{GOOD}$*], [I* $_{BUYER}$*]* ***purchased*** *[from a department store* $_{SELLER}$*] [near my apartment* $_{PLACE}$*].*

Outros elementos podem também aparecer neste cenário, como Place (local onde a transação foi realizada) e Time (momento em que o acordo foi feito). Além disso,

alguns verbos estão semânticamente ligados a este Frame, como é o caso das palavras buy e purchase. Todos esses elementos são classificados e melhor descritos na seção que segue.

2.3.1 Elementos de Frame e Unidades Lexicais

Os contituintes do Frame são formalizados em duas categorias: as unidades lexicais (UL) e os elementos de frame (EF). As unidades lexicais evocam o cenário de fundo designado pelo Frame. São constituidas pelo par palavra-significado. Cada uma das diferentes valências de uma UL irá evocar diferentes Frames. No exemplo da sentença anterior, a palavra purchased trouxe a tona a ideia de compra de algum bem.

Os elementos de frame funcionam como argumentos da unidade lexical, completando a ideia evocada por essa palavra. Na sentença de exemplo, os elementos assumem os papéis de good, buyer, seller e place, completando o significado apresentado pela palavra purchased.

Os Elementos de Frame são classificados em termos de quanto eles são centrais para um Frame. Existem três níveis, a saber [Baker et al., 1998]: (i) nuclear, (ii) periférico e (iii) extra-temático. Os elementos nucleares representam componentes que expressam algum conceito necessário a um Frame, fazendo aquele Frame único e distinto em relação aos outros Frames. O Frame Commerce_buy, por exemplo, tem como elementos nucleares os FEs Buyer e Goods. Ficaria difícil tentar imaginar um cenário de compra em que não haja um comprador e um bem. Um elemento nuclear, mesmo quando não está explicitamente presente, recebe uma interpretação e é classificado como um instanciação nula (que será abordado mais à frente). Por exemplo, o verbo arrive, usado na sentença “John arrived”, não possui o elemento Goal, ou o destino. No entanto, o local onde John chegou pode ser captado de alguma forma no contexto onde esta sentença está inserida.

Os elementos periféricos, por sua vez, não introduzem nenhuma informação independente ou que torne o Frame único. Eles podem ou não ocorrer em um frame, como o caso de elementos que caracterizam localização, tempo, modo, etc. Os elementos

extra-temáticos não pertencem conceitualmente ao frame em questão. Eles pertencem, na verdade, à frames abstratos.

Como mencionado anteriormente, os elementos de frame nucleares sempre estão presentes no frame, mesmo que implicitamente. Quando é este o caso, dizemos que se trata de uma instanciação nula. No exemplo abaixo, o elemento de frame Goods não esta presente na sentença, mas como é um EF nuclear, é marcado como uma instanciação nula:

[He $_{BUYER}$*]* ***bought*** *[wisely* $_{MANNER}$*] and he bought adventurously. [DNI* $_{GOODS}$*]*

Três tipos de instanciação nula são reconhecidos, a saber: (i) Instanciação Nula Definida (Definite Null Instantiation - DNI), Instanciação Nula Indefinida (Indefinite Null Instantiation - INI) e (iii) Instanciação Nula Estrutural (Constructional Null Instantiation - CNI). Os tipos DNI são usados em situações em que o elemento não está presente mas pode ser entendido pelo contexto de onde o texto está inserido. O segundo caso, INI, são usados quando os objetos de verbos não estão presentes, como é o caso dos verbos comer e beber. Nem sempre se vê um complemento apesar deles necessitarem sintaticamente de um objeto. As instanciações CNI se tratam das omissões permitidas pela construção gramatical, como a omissão do sujeito em sentenças imperativas.

2.3.2 Relações entre Frames Semânticos

Os Frames Semânticos não existem de forma isolada. Os Frames possuem relações uns com os outros, formando uma rede. Por exemplo, um cenário expressado através de uma sentença que denota a compra de um produto. Toda vez que alguém compra algo, uma outra parte esta vendendo. Assim, uma sentença que denota uma situação de compra de produto pode ser expressada em uma outra sentença com a venda deste mesmo produto. Porém, com papeís trocados, o comprador em um é o vendedor em outro, e vice-versa. Logo, precebe-se que esses dois Frames não são totalmente isolados, existe uma ligação entre eles.

O uso de relações pode trazer diversos benefícios para a criação de Frames. Um

desses benefícios seria o de aumentar a compreensão dos Frames e seus elementos. Um Frame complexo pode ser melhor detalhado quando relacionado à um Frame já existente, mais simples. Pode-se também aumentar a robustez desta rede de Frames uma vez que podemos associar diferentes Frames, criado por pessoas diferentes, diminuindo a possibilidade de classificações divergentes. Ao todo são 8 tipos de relações entre Frames: (i) Inheritance, (ii) Perspective on, (iii) Subframe, (iv) Precedes, (v) Inchoative of, (vi) Causative of, (vii) Using e (viii) See also.

A relação mais forte delas é a Inheritance, que corresponde que um Frame é um tipo mais específico de um Frame mais geral. Todas as propriedades que valem para o pai valem também para o Frame descendente. O Frame Commerce_Buy herda do Frame Getting, já que comprar algo é um caso mais específco de obter a posse de algo. Neste caso, existe uma correspondência entre os elementos dos dois Frames, como mostrado na Figura 2.2.

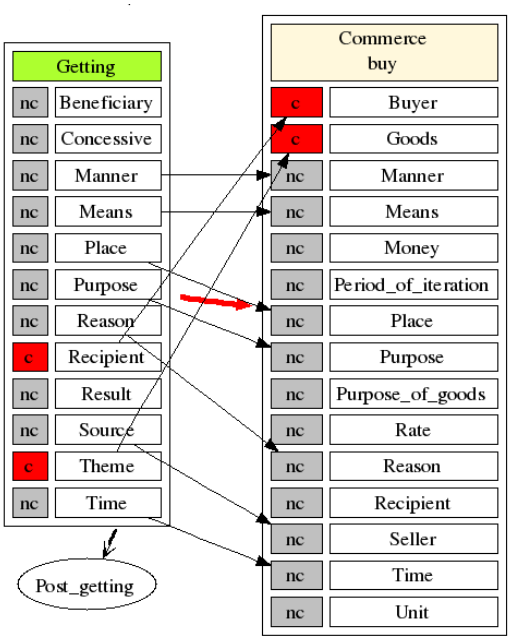

Figura 2.2: Relação de herança entre Frames.

A relação Perspective On indica mais de um ponto de vista para um certo cenário. Cada um desses pontos de vista representa um Frame diferente e cada um deles se relaciona com um Frame neutro. Continuando com o exemplo do Commerce_buy, ele

está em perspectiva como o Frame neutro Commerce_Goods_Transer. Este Frame, por sua vez, também está em perspectiva como o Frame Coomerce_Sell. Logo, podemos entender que os Frames de compra e venda reprentam pontos de vistas diferentes de um mesmo evento, a transferência comercial de bens.

Alguns Frames apresentam uma alta complexidade de componentes, estados e transições que se torna necessário separá-los em partes mais simples. Essas parte fazem o papel de Subframe em relação ao Frame mais complexo. Neste caso, os elementos do Frame pai são mapeados para os diversos subframes. No caso do Frame Criminal_Process, vemos que ele possui vários subframes, correspondendo à cada uma das diversas fases de um processo criminal. Pode-se ver também que o próprio Frame Criminal_Process é um subframe de Criminal_Scenario. Todas estas relações são ilustradas na Figura 2.3. Também nesta figura podemos ver uma relação entre os subframes de um Frame Complexo: Precedes. Os arcos pretos mostram a sequência de estados de eventos de cada um dos subframes. O ato do crime precede uma investigação criminal que, por sua vez, precede um processo criminal.

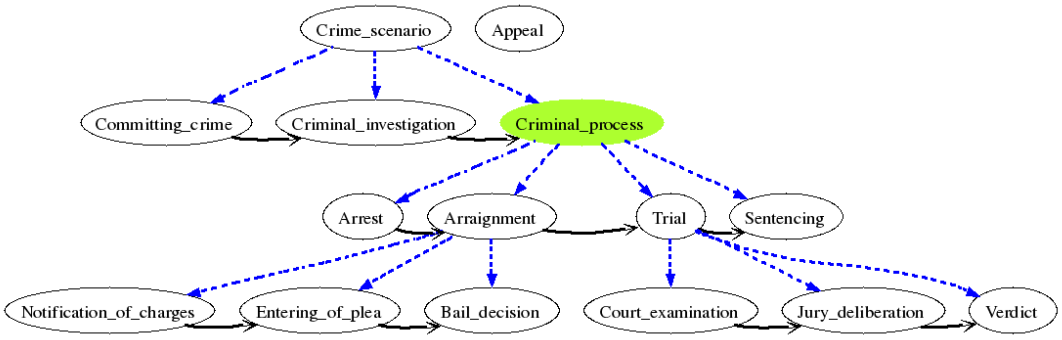

Figura 2.3: Frames e Subframes.

A relação Causative of denota situações em que um evento se torna o motivo para a ocorrência de outro. Na Figura 2.4, está ilustrado a ligação (em amarelo claro) entre os Frames Giving e Getting. A relação Inchoative of é usada entre Frames cuja ocorrência de um inicia a existência do segundo. Na mesma figura, o Frame Getting inicia uma situação de possessão, expressada pelo Frame Possession (em amarelo escuro).

A relação Using pode ser facilmente confundida com a relação Perspective on. No

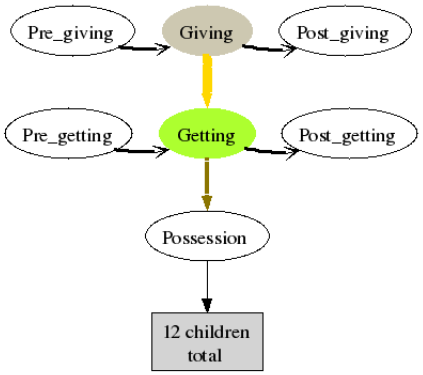

Figura 2.4: As relações Causative of e Inchoative of.

entando, essa é usada em relação com Frames mais abstratos, onde somente parte dos elementos do Frame filho possui uma correspondência com o Frame pai. Por exemplo, o Frame Volubility usa o Frame Communication, já que a fluência descreve uma qualificação de um evento de comunicação. Por fim, a relação See Also é usada entre grupos de Frames muito similares, onde suas diferenças devem ser muito bem comparadas e contrastadas.

2.4 Redes Bayesianas

Variações de modelos probabilísticos baseados em grafos acíclicos direcionados (GAD) foram primeiramente utilizados em diversas áreas, como genética, ciência cognitiva e inteligência artifical [Pearl and Russel, 2000]. Mais tarde, esses modelos passaram a se chamar Redes Bayesianas. Seu desenvolvimento, iniciado no final dos anos 70, foi motivado pela necessidade de um modelo combinatório de evidências que pudesse ser analizado tando de cima para baixo (top-down) como de baixo para cima (bottom-up), respectivamente nos âmbitos semântico e perceptivo. A capacidade das Redes Bayesianas de fazer inferências bidirecionais combinados com uma base probabilística rigorosa fizeram com que fossem amplamente utilizadas em problemas que envolvessem raciocínio em ambientes com incerteza, substituindo outros métodos *ad hoc* baseados em sistemas de regras [Heckerman et al., 1995; Jensen, 1996].

Apesar do rigor matemático, as Redes Bayesians são intuitivas uma vez que são

representadas como grafos. Os nós são variáveis aleatórias e os arcos representam as dependências probabilísticas entre essas variáveis [Ben-Gal, 2007]. As probabilidades são estimadas através de métodos computacionais e estatísticos. Logo, um arco saindo de um certo nó A até um certo nó B, significa que a variável A "influencia" a variável B.

O fato da rede ser um grafo acíclico garante que nenhum nó possa ser seu próprio antecessor ou seu próprio descendente. Essa caracteristica é importante para a fatoração da probabilidade conjunta das variáveis. Outra característica importante é que, mesmo que a dependência seja representada por arcos direcionados, o processo de inferência sobre uma Rede Bayesiana opera propagando a informação de forma bidirecional [Pearl and Russel, 2000]. Isso se deve, em parte, ao teorema de Bayes, que calcula a probabilidade de ocorrência de um evento dado um segundo evento em termos da probabilidade da ocorrências do segundo evento dado o primeiro:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Além do grafo, que representa a parte qualitativa do modelo, temos também as tabelas de distribuição de probabilidade condicional, parte quantitativa do modelo. Cada nó, ou variável, possui um dessas tabelas, disponibilizando as probabilidades da ocorrência de tal variável dadas suas dependências, ou os nós com arcos entrantes. A Figura 2.5 mostra um exemplo de Rede Baeysiana e será mais explorado à frente. Mas é importante observar as tabelas de probabilidade de distribuição condicional em cada nó. O tamanho de cada tabela é determinado pelo número de depêndencias.

De forma genérica, uma Rede Bayesiana B pode ser definida como pelo par $B = \langle G,\Theta \rangle$ [Ben-Gal, 2007], onde G é um o grafo acíclico direcionado e seus nós $X_1, X_2, ..., X_n$ representam as variáveis e os arcos, dependênca entre as variáveis. O componente Θ codifica os parâmetros da rede, onde $\Theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ para cada caso de x_i de X_i condicionado por π_i. Desta forma B define unicamente a distribuição de probabilidade conjunta sobre estas variáveis, como:

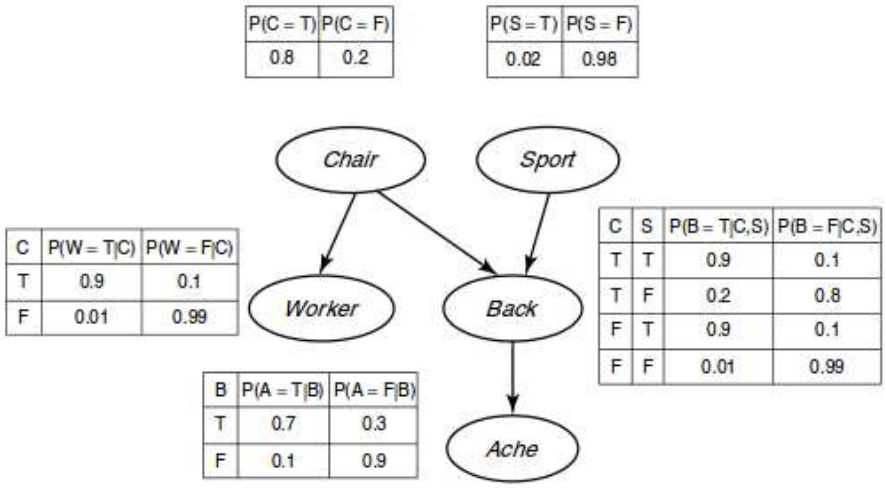

Figura 2.5: Exemplo de Rede Bayesiana. Fonte: [Ben-Gal, 2007].

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \Theta_{X_i|\pi_i} \tag{2.2}$$

A Figura 2.5 mostra um exemplo simples de Rede Bayesiana. Geralmente, possuimos algumas varíaveis conhecidas, que são tidas como observadas ou evidências; e temos variáveis em que seu valor não pode ser percebido, também chamada de hipótese, cuja probabilidade desejamos calcular. Neste exemplo, esta variável seria Back (B), que representa algum problema nas costas. O problema nas costas gera a dor, representado pela variável Ache (A). Por isso, o arco saindo de B e chegando em A. O problema nas costas pode ser resultado de dois fatores: atividade esportiva praticada de forma indevida, Sport (S); e uso de cadeiras impróprias no trabalho, Chair (C). A variável C também influencia outra variável, Worker (W), que representa o fato de algum colega de trabalho também ter problema nas costas. Todas as variáveis são binárias, podendo assumir os valores Verdadeiro (T) ou Falso (F).

Continuando com o exemplo, poderiamos calcular a probabilidade da pessoa usar cadeiras impróprias no trabalho (C = T), dado que esta mesma pessoa apresente dor nas costas (A = T):

$$P(C = T|A = T) = \frac{P(C = T,A = T)}{P(A = T)}$$

sendo que:

$$P(C = T,A = T) = \sum_{S,W,B \in \{T,F\}} P(C = T)P(S)P(W|C = T)P(B|S,C = T)P(A = T|B)$$

e que:

$$P(A = T) = \sum_{S,W,B,C \in \{T,F\}} P(C)P(S)P(W|C)P(B|S,C)P(A = T|B)$$

Repare que se usassemos uma tabela com todas as variáveis, o tamanho poderia crescer muito (complexidade $O(2^n)$, com n sendo o número de variáveis). Mas a complexidade de se somar todas as tabelas de probabilidade é de $O(n)$.

Capítulo 3

Artigos

Como foi mencionado na introdução, este capítulo apresenta os três artigos produzidos durante esta pesquisa. O primeiro artigo, intitulado *Semi-automatic Follow-up of Graduates*, foi publicado nos anais da XXXI *International Conference of the Chilean Computer Science Society* (SCCC 2012). O segundo artigo, intitulado *Applying ontology to align natural language sentences and frames*, ainda não foi submetido. O terceiro artigo, intitulado *Acompanhamento da Evolução Profissional de Egressos com Sistemas Multiagentes*, foi enviado para avaliação à Revista para o Processamento Automático das Línguas Ibéricas (Linguamática).

3.1 Artigo I: Semi-automatic Follow-up of Graduates

Diego Fialho Rodrigues, Alcione de Paiva Oliveira,
Jugurta Lisboa Filho e Alexandra Moreira

In: XXXI International Conference of the Chilean Computer Science Society (SCCC 2012). 2012, Chile.

Abstract

Many institutions need to gather and store information about people connected to their activities. An automated system to accomplish this task needs to overcome several challenges: collect information on textual and semi-structured textual databases, execute textual inferences, identify individuals, integrate distributed information, etc. Multiagent systems may help solve these problems gradually and scalably. This article describes the design and implementation of a Multiagent system that collects public information about graduates gathered from a mailing list.

Keywords: component, multiagent, ontology, Follow-up of people, frames.

3.1.1 Introduction

Institutions need to find and store information about people connected to their activities. For example, educational institutions need to monitor the professional development of its graduates; commerce companies need to monitor the behavior of their customers, media companies need to collect information about the actions of public persons, etc. On the other hand, there is a vast amount of information about individuals publicly available in several formats and accessed in a non-structured fashion: social networks, electronic journals, resume databases, general sites, search engines, etc. Due to this diversity of sources and formats it is impossible to manually maintain a database of updated information about people related to an institution. However, an automated system to accomplish this task needs to overcome several challenges: to collect information on

textual and semi-structured textual databases, execute textual inferences, identify individuals, integrate distributed information, etc. Multiagent systems (MAS) [Wooldridge, 2002] technology is presented as an alternative to solve these problems gradually and scalably. MAS allows the construction of systems that can operate distributed and decentralized, allowing the addition of processing elements transparently.

Our main goal is to make the tracking of graduates less laborious by providing means for this task to be performed in a semi-automatic way. To do this we use semantic frames, natural language processing techniques, reasoning over ontologies, and search and storage of data in remote repositories. In order to make the automatic classification of the results, we use Bayesian networks.

This article describes the design and implementation of a MAS that collects and stores information on graduates that can be found in public repositories. It is not our intention to leave the system closed for future modifications. New functionality can be added using different approaches. The use of Multiagent System helps in this regard.

The paper is structured as follows. The next section presents recent studies in the tracking people field. Section 3.1.3 introduces the concept of Multiagent System and shows why we choose this kind of architecture. Section 3.1.4 shows the details of the model which was constructed over a Multiagent architeture. Section 3.1.4.1 shows one of the semantic frames utilized in this work. Section 3.1.4.2 presents the ontology constructed over the ontologies DUL and Time. Section 3.1.4.3 presents the ontology created to formalize the frames. Section 3.1.4.4 shows how bayesians networks were applied to rank the frames. Next, in section 3.1.5, we present a case study along with an example. Section 3.1.6 reports on related work. Finally, section 3.1.7 presents the conclusions of the research carried out.

3.1.2 Tracking People

People and institutions of the modern world live a dilemma: there has never been so much information available and ready to be accessed but paradoxically, never was so little time to process it. Nevertheless, it is known that the proper management

of information is critical to the success of individuals and organizations. For this to occur we must rely on automated systems. Automated systems that are able to collect, aggregate and manage the information available would be highly useful. Among the potentially useful systems, systems for collecting information about people would be very useful for organizations such as media companies, information agencies, educational institutions and commercial enterprises in general. Several tasks are associated with the general task of searching for people: extract people, identification of social networks [Khan and Khan, 2009], creating biographies, specialty association, association of persons to documents [Balog, 2008; Balog and de Rijke, 2008], etc.

One of the evidences of the importance of such systems is that the search for people was one of the tasks of the Fourth International Workshop on Semantic Evaluations [SemEval, 2007], organized by the Association for Computational Linguistics (ACL). A task to search for specialists was introduced in The Fourteenth Text Retrieval Conference - TREC 2005 [Craswell and de Vries, 2006]. Several researchers work on problems related to the area. Elmacioglu et al. [2007] presented a system to eliminate ambiguity in Web searches related to people's names through clustering. Artiles et al. [2007] presents criteria for evaluating the performance of systems that people seek on the Web. Popescu and Magnini [2009] presented ways to alleviate the problem of establishing erroneous co-reference in the search for people on the Web. Balog et al. [2009] also attacked the problem of name resolution in documents through clustering.

Most studies dealing with the association of persons to documents use document clustering techniques to eliminate ambiguities. The basic assumption of this approach is that similar documents tend to represent the same person. Some of the main clustering methods are [Balog et al., 2009]: Single Pass, k-Means, agglomerative clustering, and together with probabilistic latent semantic analysis (PLSA). The first three methods provide different variations of the traditional clustering methods (variation in terms of efficiency and quality). These methods are also based on the fact that different documents have the same terms when associated with the same individual. The last method is not based on this hypothesis and uses a probabilistic mathematical model

to establish conditional latent semantic relations, formally defined as:

$$p\,(t,d) = p\,(d) \sum_{z} p\,(t|z)\,p\,(z|d) \tag{3.1}$$

Where $p(t,d)$ is the probability of a term t and a document d will co-occur. $p(t|z)$ is the probability of a term t occur in a given topic z. $p(z|d)$ is the probability of a topic z occur in a document d.

All these techniques have their strengths and weaknesses and can potentially be applied in this project. But this project aims to establish a general framework for housing systems for collecting information about people and being able to use various techniques, focusing mainly on those that are based on the particular sources of information. Particularly, this paper discusses an application for a system that collects information about persons: follow-up of graduates of educational institutions. The monitoring of graduates is of great importance to educational institutions. The professional success of graduates from these institutions can be attributed in part to the training received from them. One of the criteria established by the Brazilian Ministry of Education to evaluate the educational institutions is the monitoring of its graduates. However, this is an almost impossible task to be performed in its entirely. It is necessary to send mail from time to time to possibly outdated addresses, asking the alumni to update their data and addresses. Hence, an automatic system to seek information from various sources about the graduates would be useful. Obviously, for and ethical privacy reasons these systems should only collect information from public repositories, available directly or with the consent of the parties involved.

3.1.3 Multiagent Systems

Multiagent System (MAS) is an area within the computer science that deals with aspects of distributed computing systems applied to artificial intelligence [Bordini et al., 2001]. This area arose from observations made in some natural systems, such as ant colonies, where we can see an intelligent behavior from the interaction of its elements

[Bolzan and Giraffa, 2002; Hübner et al., 2004]. The research on MAS escapes from traditional paradigms, for it has as its object of study the collectivity rather than the individual. In the field of Multiagent Systems, it is studied the behavior of an organized group of autonomous agents, which cooperate to solving problems that are beyond individual capacities of resolution [Bolzan and Giraffa, 2002; Hübner et al., 2004].

The most used definition for agent is the one presented by Wooldridge et al. [1995]. According to them, an agent is a computer system situated in some environment, which is able to take autonomous actions in this environment to achieve its goals. Moreover, to achieve their goals the agents can interact with each other through cooperation and coordination. The communication can be done through the exchange of messages.

Multiagent Systems have become a powerful tool for solving IT problems, which are increasingly complex. Agent-oriented software development offers autonomy to system components, flexibility, and a way to apply artificial intelligence to specific parts of the software.

The construction of automated systems capable of gathering information from various sources in various formats and even in natural language texts is not a trivial task. The system should be able to act in a distributed and collaborative way and to be able to make inferences and pattern matching. These characteristics are associated with artificial intelligence systems, particularly multiagent systems.

The system should also have the freedom to incorporate new approaches to solve the problem without causing major impacts on already existing features. With MAS, this could be done naturally with the addition of new agents. From the standpoint of performance, the various agents could work in parallel taking advantage of the inborn capabilities offered by Multiagent architectures.

3.1.4 The Model

The proposed model uses MAS architecture for collecting and storing information about graduates. The monitoring activity involves agents performing related but distributed tasks. There are repositories of data scattered in different locations and the information

contained therein need to be matched and processed.

The model has six agents, namely: (i) University Web Server, (ii) Email Reader, (iii) Questionnaire Sender, (iv) Questionnaire Answers Reader, (v) Database Manager, and (vi) NL Processor (Natural Language Processor). The first one is responsible for seeking information of graduates in a web service of the educational institution. The query is made to the web server every 6 months, period in which new students graduate and leave the university. The Email Reader has the function of reading the mailing list of alumni. It can contain information about the career paths of graduates.

The Questionnaire Sender agent has the function of sending questionnaires to the mailing list of former students with questions about their careers. Students fill out the forms and the answers are sent automatically to the data repository, but the information is not correctly structured yet. The Questionnaire Answer Reader, in turn, is instructed to read the questionnaire responses and write the data to the repository in a structured manner. There may be conflicting data and this agent is responsible for preventing this.

The Database Manager coordinates a data repository containing the registration of students, cities, businesses, offices, functions, etc. Any read or write operation on the repository is mediated by this agent.

Finally, the NL Processor aims to process the messages in the mailing list and, using the information provided by the other agents, it tries to find evidences linking the graduates and their companies. Only the fourth agent will be described in this paper.

The Fig. 3.1 shows the dependence between the agents through an actor diagram. The actor diagram is part of the Tropos Methodology, used for modeling multiagent systems. Further information about Tropos can be found in Giunchiglia et al. [2002].

The messages contained in email lists are written in natural language and it is not an easy task for a computer to understand it. Thus, it becomes necessary to use NLP techniques so that the machine can make inferences about the collected information. To do this we used semantic frames [Fillmore, 1977] supported by a domain ontology. Finally, Bayesian networks are used to numerically rank the evidences found as suggested

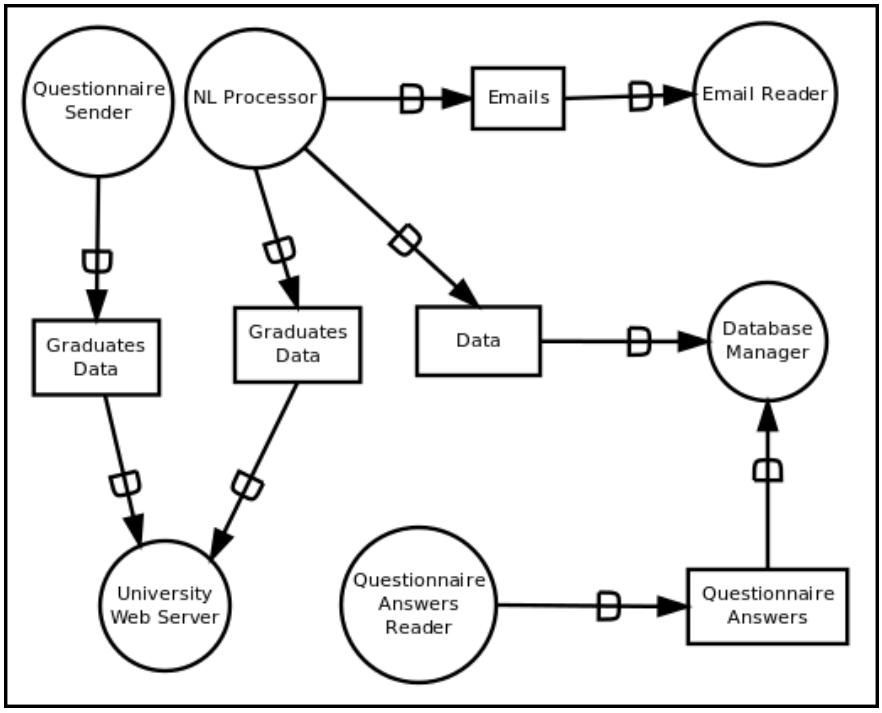

Figure 3.1: Dependence between agents.

by Moreira [2012].

Semantic Frames

Semantic Frames are conceptual structures that describe a particular type of situation, object, or event along with its participants [Baker et al., 1998]. This participants are called *Frame Elements* (FEs). The Frame Elements are divided into two groups: the Core and non-Core. The core elements always occur, implicitly or explicitly. The non-core elements may eventually occur. Also, each frame is evoked by a *Lexical Unit* (LU), which is pairing of a word and a meaning. The *Frame Elements* act as arguments for the *lexical units*, completing the sense presented by the word.

The presence of certain words in a text can reveal the occurrence of a frame and thus the context defined by it. Therefore, frames can be used to identify sentences that denote the idea of connection between graduates and enterprises where they work.

The frames creation was done through the analysis of a *corpus* obtained from the alumni mailing list. It has about 5,000 messages that were sent between 1999 and 2011.

By reading the mailing list, we noted that it is common for a graduate to offer jobs to the company where he works. Along with the job offer, the graduate can mention other important information, like the city where the company is located, the date he/she joined the company, the function exercised, etc. Usually the sentence is in first person and the company name is mentioned.

In the example bellow, the word "trabalho" acts as the lexical unit, evoking the Frame. Also, there are others participants: COMPANY, "X" (omitted), GRADUATE, "eu", JOB OPPORTUNITY, "está com uma vaga", and PLACE, "São Paulo".

A [X COMPANY], empresa onde [eu GRADUATE] ***trabalho****, [está com uma vaga JOB OPPORTUNITY] para Desenvolvedor Java em [São Paulo PLACE].*

(The [X COMPANY], the company where [I GRADUATE] ***work****, [has a vacancy JOB OPPORTUNITY] for Java Developer at [Sao Paulo PLACE].)*

All these elements are present in one of the frames obtained and can be seen in Fig 3.2. We used the same notation adopted in the Berkeley FrameNet. The Lexical Units are shown in black background and the Frame Elements are shown in a colored background. Along with the description of the components of the frame, are shown also examples of occurrence of the components found in the corpus. The language from which the frame was built is Brazilian Portuguese.

Other frames were identified during the reading of messages. Also, new frames can be identified in other data sources. However, we will use only this frame in this article, since the process is similar to other frames.

The Domain Ontology

Inference over natural language requires the use of semantic and syntactic formalization combined with the world knowledge offered by ontologies [Scheffczyk et al., 2006b]. Semantic frames can be used for semantic analysis of sentences but not for reasoning about the nature of the elements contained therein. Ontologies can be used to formalize the elements of the domain in question and to create a link between their entities and

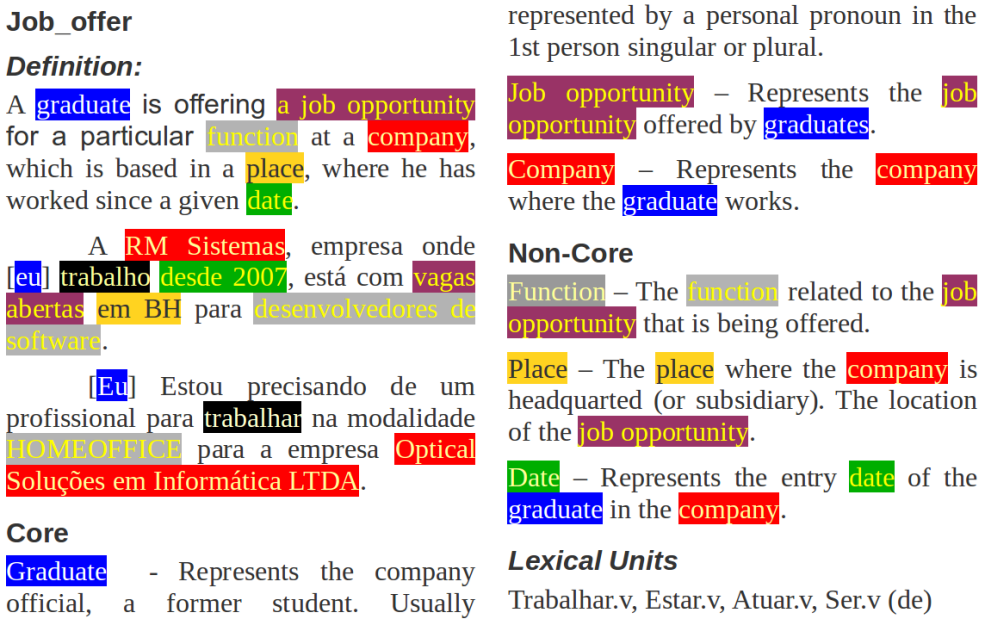

Figure 3.2: The Job_Offer Frame.

the semantic types of the Frame Elements [Carrasco et al., 2011]. During the parsing process it is necessary to know whether a piece of text is filler for a certain Frame Element [Scheffczyk et al., 2006a]. This can be done by matching the semantic type of the filler with the classification given by the ontology. For example, the piece of text “desde 2006” (since 2006) can be classified as an instance of the “Open Interval” class of the ontology, which in turn is a kind of “Interval”. The entity “Interval” in the ontology can be mapped to the Frame Element “Date” in the Job_Offer Frame.

The ontology that we create has the purpose of describing the elements that occur in the context of the employment relationship between graduates and enterprises. The ontology was built over the ontologies DUL[1] and Time[2]. The ontology describes events as “Exercise of an office” and “Job offer”, organizations such as “Company” and “University”, places like “City”, “State” and “Country”, and so on. To classify the different types of time intervals, some classes were also created over the Time ontology, namely: “Closed Interval”, “Indefinite Interval”, and “Open Interval”. Some entities were reused from the underlying ontologies, e.g. “Role”.

[1] www.loa.istc.cnr.it/ontologies/DUL.owl
[2] raw.github.com/RinkeHoekstra/lkif-core/master/time.owl

There was no need to create many properties. Most of classes use existing properties, like the relation “is part of” between “Department” and “Organization”. The Ontology is shown in Fig. 3.3. The parts (a), (b), and (c) contain the classes while the parts (d) and (e) show the properties. The elements we created are in boldface.

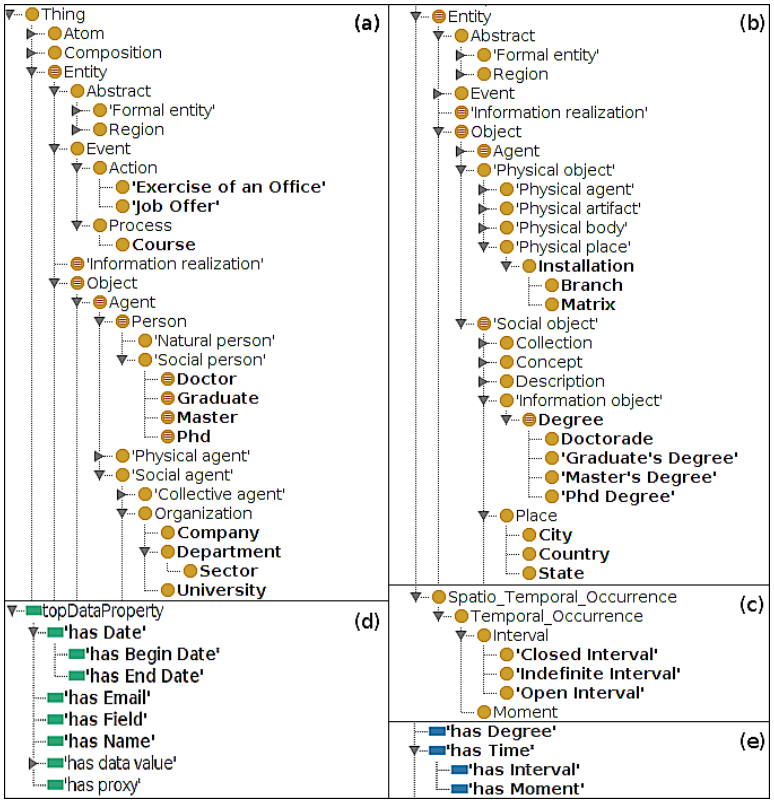

Figure 3.3: The Domain Ontology merged with DUL and Time. A, B, and C: the classes. D and E: the properties.

Semantic Frame Ontology

In order to recognize the context of an employment relationship between the companies and the graduates in a text, it is necessary that the semantic frames are properly identified. This is done through the identification of the components of the frame (Frame Elements and Lexical Units) in chunks of text in the sentence. Frames provide the means for semantic and syntactic analysis of natural language. However, this analysis is done by humans and not by computers. Thus, the formalization of frames somehow becomes necessary in order to make the parsing of text automatically.

On the other hand we have the domain ontology, which formalizes the entities in question and can be used for reasoning. But it has no suitable linguistic components. To make the connection between the domain ontology and the semantic frames, we use a new ontology: an ontology that describes the semantic frames domain. This ontology was based on the work of Scheffczyk et al. [2006a].

Fig. 3.4 shows a simplified excerpt of the Semantic Frame Ontology. The class "Syntax" represents the chunks of text. It has the relations "evokes" and "fillerOf" with the classes "Frame" and "FE", respectively. A Frame also has several FE, which is represented by the relation "hasFE". Besides the inheritance relationship, naturally represented in the ontology, there are other relations between frames, namely: (i) Perspective on, (ii) Subframe, (iii) Precedes, (iv) Inchoative of, (v) Causative of, (vi) Using, and (vii) See also.

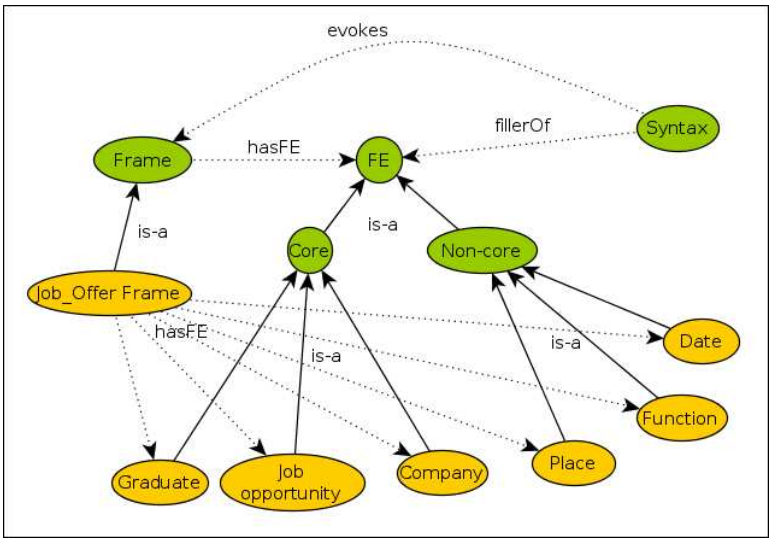

Figure 3.4: The Semantic Frame Ontology.

Each Frame Element has a Semantic Type (ST). Thus, the connection between frames and the domain ontology is made through the relationship "hasST" between a FE and one of the classes in the domain ontology. For example, the "Job Opportunity" FE has the semantic type equals to "Job Offer". Hence, there is a relation "hasST" between these two elements.

Bayesian Networks

Once we find sentences that conform to our semantic frames, it becomes necessary to rank the text to see how that evidence is strong. Balog et al. [2006] used the Bayes' theorem to determine the probability of a candidate given a document. However, our work focuses on the marriage of sentences and not entire documents. Meurs et al. [2009] used a model with dynamic Bayesian networks taking in consideration Semantic Frames where each word represents a time slice.

In our study, we use Bayesian networks to classify the sentences. According to Friedman et al. [1997], bayesian networks are directed acyclic graphs that encodes the joint probability distribution over a set of random variables. Each variable is represented by a vertex and the correlation between the variables are represented by the edges. For each variable, there is a table of probabilistic values representing the local conditional distribution given its parents. The model was constructed based on the semantic frames of section 3.1.4.1.

Frames are evoked by lexical units [Ruppenhofer et al., 2010]. The lexical units in turn, evokes the frame elements. That is, the frame elements act as fillers for the lexical units. Based on these affirmations, we created a Bayesian network over the Job_Offer Frame. The network was built using the Weka software [Hall et al., 2009]. A list of sentences were manualy annotated and then classified as a ocurrence or not of the frame. From all the algorithms avaiable in the software, the Tree Augmented Naive Bayes (TAN) [Friedman et al., 1997] showed the best result (94.0% of cases evaluated correctly). The network is illustrated in Fig. 3.5. The vertices represent each of the elements of the frame, the lexical unity, and the frame itself.

The behavior of our model is always to seek sentences that contain the lexical units of our semantic frames. Once the sentence is found, the task is to find the Frame Elements. Then, the sentence is ranked according to the presence or absence of the elements of our frame. The more elements are found in the sentence the more likely it being classified as a instance of the Job_Offer frame.

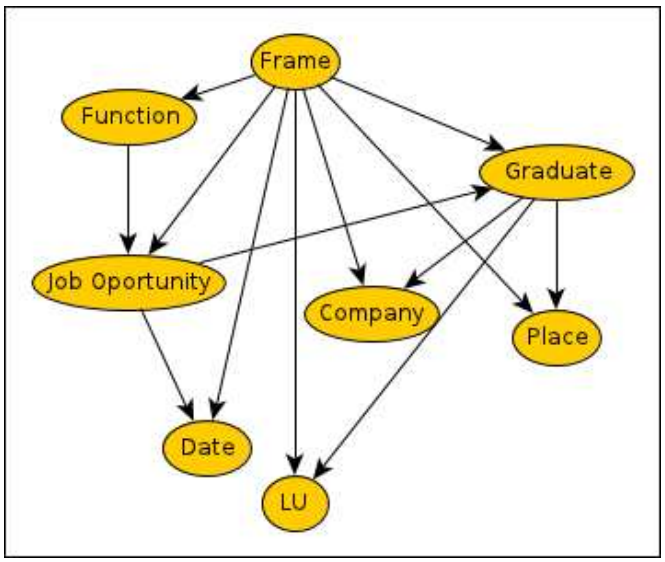

Figure 3.5: The Bayesian Network over the Job_Offer Semantic Frame.

3.1.5 Case Study

A multiagent system will be used for recognition of the frames present in the e-mails. Each frame is represented by a specific agent. Thus, as new frames are identified, new agents can be added to the system. Besides the semantic frames, the agent also has an ontology, which is global, and a specific Bayesian Network.

First, the Mail Reader agent will get the messages sent by the graduates on the list of e-mails. Next, each message will be broken up into a collection sentences. These sentences are then ready to be processed by the NL Processor agents, which are related to the frames.

The NL Processor agents scan the sentences looking for statements that demonstrate the occurrence of the semantic frame. First, the system filters the sentences that contain any of the lexical units of the frame. Once a lexical unit is found, the agent tries to find the frame elements. The ontology is used to discover the nature of each part of the sentence and decide whether or not it is a frame element. Once we know which FEs are present and the ones that are absent, the probability that the text is a case of a frame is calculated through the Bayesian network.

Here is a example of a sentence annotaded with the element of the Job_Offer Frame:

*A empresa onde [eu $_{GRADUATE}$] **trabalho** esta com algumas [vagas em aberto $_{JOB\ OPPORTUNITY}$], se alguem estiver afim de trabalhar em [X $_{PLACE}$] a empresa é [Y $_{COMPANY}$].*

*(The company where [I $_{GRADUATE}$] **work** has a few [job vacancies $_{JOB\ OPPORTUNITY}$], if someone wants to work in [X $_{PLACE}$] the company is [Y $_{COMPANY}$].)*

The Table 3.1 shows the processing result of the sentence. Place and company were omitted. Some elements are classified at ontological level and a mapping is done to the corresponding frame element.

Table 3.1: Example of a frame matching

Sentence	Ontology	Frame
I	Graduate	Graduate
work	Exercise of an Office	Lexical Unit
job vacancies	Job Offer	Job Opportunity
X	City	Place
Y	Company	Company

Once the information has been classified, then it is possible to rank the sentence by using the bayesian network. In the example of Table 3.1, the probability of the sentence to be an occurrence of the frame Job_Offer is 99.22%. For comparison, if the lexical unit is not mentioned, the probability drops to 39.30%.

If the sentence is marked as an instance of a frame, the extracted information is stored in the data repository by the Database Manager agent. The information of the sender of the message is also taken into consideration in cases that the graduate uses pronouns to refer to himself. The same happens with some dates. If, for example, the word found is “hoje” (today), the information will be automatically converted to the date of dispatch of the message.

The Email Reader agent execute in defined time intervals. Thus, whenever a message is sent to the list, entire parsing process is run again.

3.1.6 Related Work

As a somewhat related work we can cite the association of persons to documents [Balog and de Rijke, 2008], which analyze texts to identify people. Carrasco et al. [2011] presented a computer system for tracking wild animal's illegal trade in social networks. The FBI has the Carnivore system [McCarthy, 2001], which operates on private messages of Internet users, which leads to discussions of privacy violation.

Scheffczyk et al. [2006a] presented a way of linking large lexical resources with world knowledge via ontologies. In Scheffczyk et al. [2006b], the same author proposed a set of rules for mapping Frame Elements with elements of the SUMO[3] Ontology.

3.1.7 Conclusions

This paper proposed a multiagent system for tracking graduates. We use frames along with ontologies to search for sentences that bring evidences relating graduates and the company where they work. The Bayesian networks were used to rank the results and assist in decision making.

The monitoring of graduates is of great importance to educational institutions. A tool with the potential to help in accomplishing this task in a less laborious can be of great value. New frames can be identified and entered into the system by taking advantage of the great flexibility that multiagent systems offer. Another technique that could be added to the system would be to look for patterns characteristic of the form of communication that is being used, such as signatures at the end of emails and the use of corporate email.

Acknowledgment

This work is financed by funding agencies FAPEMIG, CNPq and FUNARBE and by the Gapso Company.

[3] http://protege.stanford.edu/ontologies/sumoOntology/sumo155.zip

3.2 Artigo II: Applying Ontology to Align Natural Language Sentences and Frames

Diego Fialho Rodrigues, Alcione de Paiva Oliveira,
Jugurta Lisboa Filho e Alexandra Moreira

Abstract

Detecting the underlying semantics of a natural language sentence is a difficult task even for human beings. The scene or context related to the sentence is what make it possible to elucidate its meaning. Therefore, establishing this relationship is a key point for natural language processing. The aim of this work is to use ontology to establish the relationship between a sentence and the scene, in order to allow the analysis of natural language sentences within the context of monitoring the professional development of alumni of an educational institution.

3.2.1 Introduction

Detecting the underlying semantics of a natural language sentence is a difficult task even for human beings. According to Fillmore [2006] the scene or context related to the sentence is what allows one to understand tis meaning. Fillmore named this scene as *frame* and developed a theory to support his vision of semantics, called *frames semantics*. Establishing relationship between a sentence and its frames is a key point for natural language processing. One possible way to do this is to annotate in advance the types of lexical items of the sentence using a domain ontology [Chandrasekaran et al., 1999] and use this annotation to check the probability of the sentence being associated with a frame. This technique was proposed in [Moreira, 2012]. Here we are interested in applying this technique to a practical problem of natural language processing: tracking people. This is a delicate issue, but we must clarify that we are not interested in obtaining confidential information from people. We are interested in using an ontology to establish the relationship between a sentence and its frame, in order to

allow the analysis of natural language sentences within the context of monitoring the professional development of alumni of an educational institution.

Currently we have the massive use of the Internet with the number of users and the amount of information circulating within the network increasing. Much of this information can be used profitably by many institutions and companies, who can use these sources to their own advantage. As an example, a sales company is interested in knowing the preference of their customers to offer products that would spark their interest. Another example are the educational institutions that need to follow the career paths of its graduates.

In order to accomplish this task automatically, computers appear as a natural solution. Many of these sources provide the information in a very specific format and in structured way, like for example web services. In other sources, however, the information is in an unstructured format. Examples are social networks and mailing lists, where people exchange messages in natural language format, such as Portuguese, English, an so on. This type of information is not as easily interpreted by a computer. For this it is necessary to use natural language processing techniques, context recognition, ambiguity resolution, among others techniques.

In this paper, semantic frames are used together with ontologies to grasp the semantics underlying a sentence. More specifically, we try to monitor the professional development of alumni of an educational institution by capturing evidences revealing the employment relationship between companies and employees. The solutions employed to accomplish this task can be adapted and used in other contexts as well, like monitoring of academic development and tracking of crimes on the Internet. This research is within the scope of the discovery of relationships in textual documents, which is a topic of much current research, as shown in a recent publication in the Journal of the Brazilian Computer Society conducted by Abreu et al. [2013].

The paper is structured as follows. The next section describes the role of ontologies in the formalization of the concepts of a domain and the establishment of its types. Section 3.2.3 shows an ontology to formalize the frames. Section 3.2.4 show how

the domain ontology can be combined with the frame ontology for natural language processing. The next section presents some tests and results achieved. Section 3.2.6 reports on related work. Finally, section 3.2.7 presents the conclusions of the research carried out.

3.2.2 The Problem and The Domain Ontology

To accomplish the task of monitoring graduates, a wide margin of information needs to be represented. We need to know the institutions where he studied, the companies through which he passed, his skills, and so on. As mentioned in the introduction, there is a vast amount of information available on the Internet. In terms of monitoring professionals, we can extract information from diverse sources on the Internet, such as social networks, databases, resumes, mailing lists, etc. One of the problems is that each of these sources provides their information in different ways from fully comprehensive and structured formats to fully incomplete and ambiguous formats.

In this problem we use ontologies as a way of structuring these different formats in a single, well-defined manner. Ontologies provide terms and relationships to represent knowledge in a given domain [Chandrasekaran et al., 1999]. This ontology was not built from scratch. We use a top-level ontology, called DUL[4], as a starting point. This way, we can enjoy a pre-existing hierarchy to classify our elements. Moreover, we can reuse existing class and several relationships. With the same purpose, one ontology to deal with temporal aspects was used: the Time ontology[5].

The ontology DUL classifies its elements below the class *Entity* and this class, in turn, divides into five categories, namely: (i) *Abstract*, (ii) *Event*, (iii) *Information realization*, (iv) *Object* e (v) *Quality*. Most classes are subclass of the entity *Object*. We have represented professionals and their roles as graduates, teachers or doctors. We also have organizations here represented by universities (such as the workplace and qualification) and companies. Each company can have multiple installations, divided into headquarter and branches. The organizations are divided into departments and

[4]www.loa.istc.cnr.it/ontologies/DUL.owl
[5]raw.github.com/RinkeHoekstra/lkif-core/master/time.owl

sectors, and still have a location in our context: city, state and country. The subclasses of *Object* are shown in Figure 3.6.

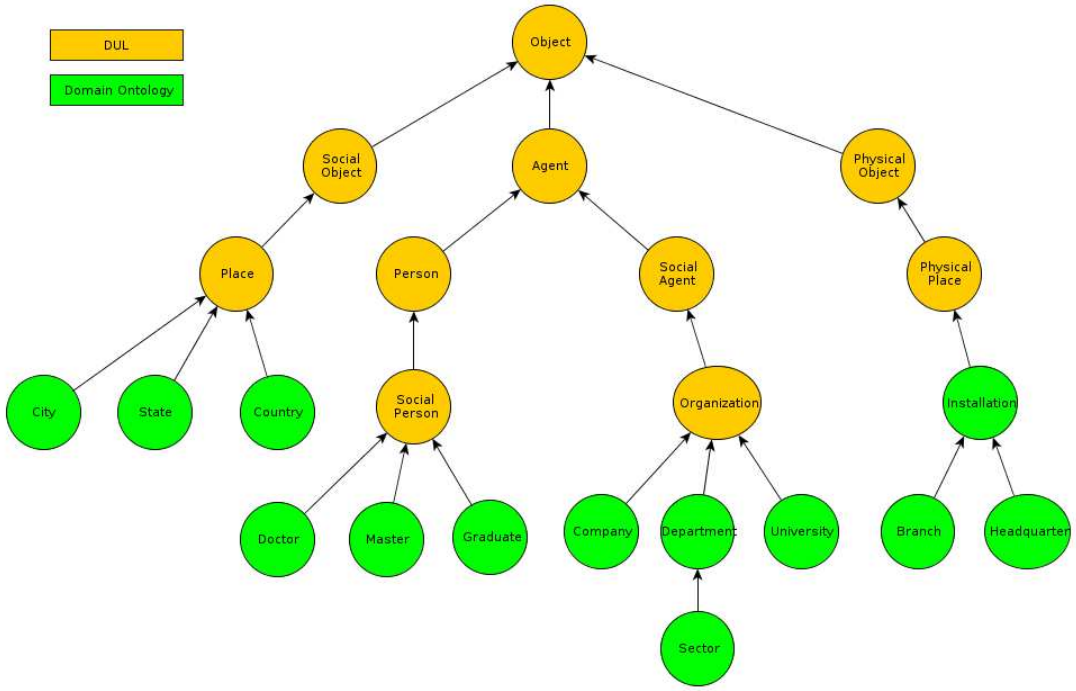

Figure 3.6: Subclasses of Object.

As sublcasses of *Event*, we have *Action* and *Process*, where are the classes *Exercise of an Office*, *Job Offer* and *Course*. These elements have relationships with people, organizations and Time Interval. For example, a person holding a position in one of the branches of a company for a certain period. The part a of the Figure 3.7 illustrates the subclasses of *Event* and part b shows the relationships of class *Exercise of an Office*.

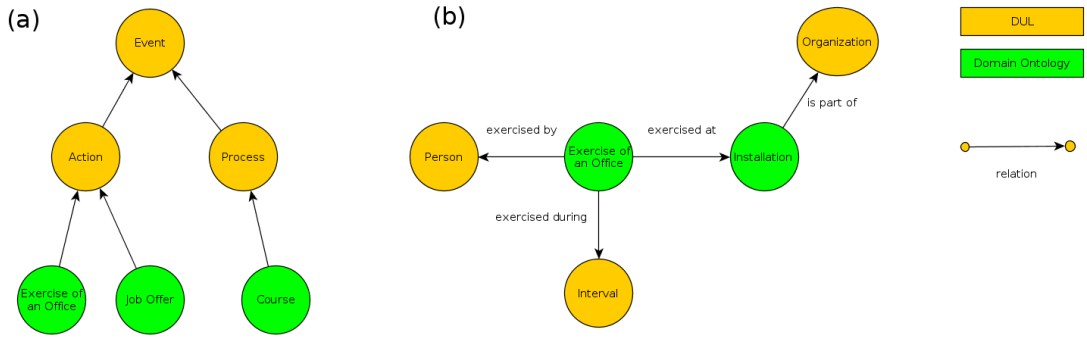

Figure 3.7: (a) Subclasses of Event. (b) Relations of Excercise of an Office.

There was no need to create many relationships since DUL ontology offers almost all the properties that were necessary. Various classes such as *Person* and *Role* were also reused. What was modeled serves the purposes of this study but it is clear that many refinements depending on the desired level of detail can be made.

3.2.3 The Semantic Frame Ontology

Semantic Frames are conceptual structures that describe a particular type of situation, object, or event along with its participants [Baker et al., 1998]. This participants are called Frame Elements (FEs). The Frame Elements are divided into two groups: the Core and non-Core. The core elements always occur, implicitly or explicitly. The non-core elements may eventually occur. Also, each frame is evoked by a Lexical Unit (LU), which is pairing of a word and a meaning. The Frame Elements act as arguments for the lexical units, completing the sense presented by the word.

In the work of Rodrigues et al. [2012], Semantic frames were used to capture job offers sent to mailing list of alumni. To extract this information to our domain ontology a second ontology was created: the semantic frames ontology. The main idea is to formalize all terms and relationships in this context to help us identify the elements in each sentence. In this work, we only work with a frame that captures only part of the context of tracking graduates. This is the job offer frame. By using this frame we can capture information such as the company where he works, the city and occupied position.

In our ontology, all concepts of frame were inserted as subclasses of *Frame_Concept*, below *Abstract* in the DUL ontology. Each frame and frame element are entered under their respective classes. There is also the Syntax entity that presents snippets of text, divided into NI (Null Instantiation - the element is implicit in the text) and Span (explicit text snippets) [Scheffczyk et al., 2006a]. The class hierarchy is shown in Figure 3.8.

Various relationships were created so that the elements are better defined. The most important are described in Table 3.2.

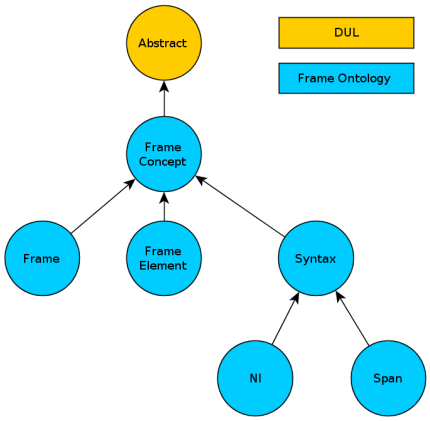

Figure 3.8: Frame Ontology Classes.

The class that represents the job offer, called *Job_Offer_Frame*, besides being a subclass of *Frame* has several properties that help define its meaning. This frame has three FEs as core, i.e. are required in the sentence. Therefore, the class has three elements of class relations *Frame_Element*. The following listing shows the relationships using Manchester syntax for OWL 2[6].

```
Subclass of:
  Frame
  hasCoreFE some Job_Offer.Company
  hasCoreFE some Job_Offer.Graduate
  hasCoreFE some Job_Offer.Job_Opportunitty
  hasFE some Job_Offer_FE
```

The non-core elements are not present because they are not mandatory. However, there is a relation with the frame in each of the frame elements, as in the case of the FE *Place*. It is also important to note the relation *hasSemanticType*. As the name says, this relation refers to the semantic type assigned to the FE. For example, the FE Place has a semantic type that points to the class *Place* in the domain ontology. In this case, it may be a city, a state or a country.

[6]http://www.w3.org/2007/OWL/wiki/ManchesterSyntax

Table 3.2: Some Object Properties For Frames.

Property	Domain	Range	Description
evokes	Span	Frame	A piece of text can evoke a certain frame, playing the role of a lexical unit
fillerOf	Syntax	Frame_Element	A piece of text plays an FE
hasFE	Frame	Frame_Element	One frame has a type of FE
hasSemanticType	Span	Thing	A piece of text has a meaning in the context of the problem
isEvokedBy	Frame	Span	A certain frame is evoked by a LU
isFEOf	Frame_Element	Frame	A FE is an element of the frame
isFilledBy	Frame_Element	Syntax	A certain FE can be represented by a piece of text

```
Subclass of:
  Job_Offer_FE
  InNonCoreFEOf some Job_Offer_Frame
  hasSemanticType some Place
```

The class *Span* represents only snippets of text and, when this text evokes a frame, it is automatically considered a lexical unit (LU class). Thus, the conditions necessary for an individual to be a subclass of LU is being Span and evoke a frame:

```
Equivalent to:
  evokes some Frame

Subclass of:
  Span
```

We are using only one frame, which captures only part of this information in the domain ontology. We can create new frames to capture other information, for

example, courses a graduate attended. The next section shows an example of the logical operations performed to verify whether or not a sentence is an instance of a Frame.

3.2.4 Using Ontologies for Natural Language Processing

Once we have the domain ontology and the semantic frames ontology together, we can make logical queries to check whether or not a sentence is the case of a frame. But for this we need to have our ontology populated with individuals, especially in the class *Span* and classes of the domain ontology. The search for frames can be summarized in 4 steps for the simplest cases[7]:

1. Find the candidates LU's for the frame in question
2. Search for LU's in the sentence
3. Search the semantic types of each FE
4. Check if the text snippets can be FE's

The first step serves to verify what are the *Spans* that have the role of lexical unit of the Frame. The *evokes* property is used, that returns a set of *Spans*. Below an example for the Frame *Job_Offer*:

```
evokes some Job_Offer_Frame
```

Once we have the list of possible *LU's*, the second step is to check whether some of these *Spans* is present in the sentence. Each *Span* has a property called *hasText* that returns the text passage it represents.

Next, we must find out what are the Frame Elements that the Frame has. This is done through the relation *isFEOf*. Furthermore, we need to know the semantic type of each of the FE's using the *isSemanticTypeOf* property (inverse of *hasSemanticType*):

```
isSemanticTypeOf some ( isFEOf some Job_Offer_Frame )
```

[7]There are more complex situations where there may be additional steps, such as checking *Coresets*

In the last step, we check whether the text snippets present in the sentence can be argument for some of the Frame Elements. This is done by searching for the text in the individuals of the class *Span*, and then checking if the text has some of the semantic types of the FE's sought in the previous step. For example, the City FE has the semantic type *Place*. Thus, we must seek *Spans* whose semantic type is equal to *Place* or any its subclasses. Below an example to verify that the words "Rio de Janeiro" can be a FE of type City:

```
Span and hasSemanticType some Place
  and hasText value ''Rio de Janeiro''
```

After all the frame elements are sought, we evaluate whether the sentence is the case of a frame. It is important to remember that the Core Frame Elements not found in the text are classified as NI (Null Instatiantion). This paper proposes to seek only the elements in the sentences through Ontology and Semantic Frames. The works of Rodrigues et al. [2012] and Carrasco et al. [2011] show ways to rank the likelihood of a sentence to be a Frame.

3.2.5 Tests and Results

To run the tests, we created a system that can identify and mark all occurrences of frame elements and lexical units referring to frame studied. The results were compared to the evaluation made by people who know the elements of the problem domain, e.g., companies, cities, dates, etc.

The evaluations of both the system and the ones made by people generates two possible results: positive (it is an element) and negative (it is not an element). These results were plotted in a receiver operating characteristics (ROC) graph [Fawcett, 2006]. The ROC curve shows the true positive rate (y-axis) and false negatives (x-axis). Thus, the nearest to the point (0, 1), the better the classification. Conversely, the results next to the diagonal line across the graph (Random Guess) represent random classifications. Figure 3.9 shows the results.

We analyzed 201 samples. Most elements obtained good results with the false

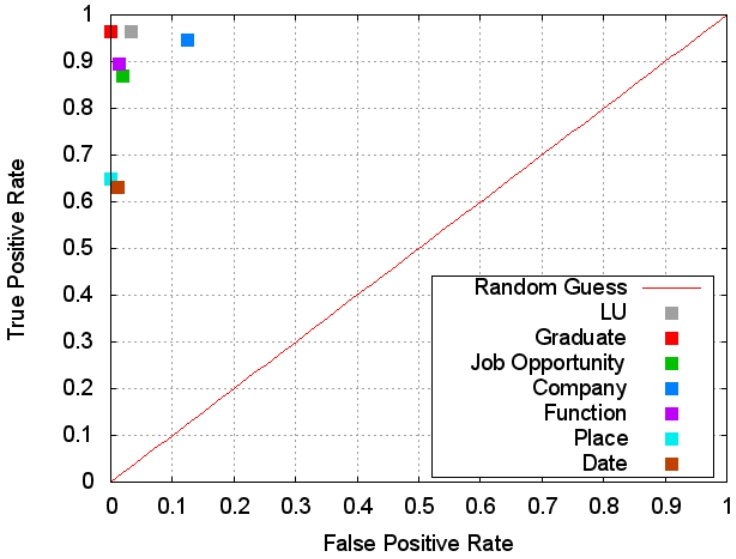

Figure 3.9: The Roc Space.

positive rate below 4 % and true positive above 85 %. The exception was the non-core frame elements Date and City. Yet the false positive rate was low (less than 1 %).

3.2.6 Related Work

As related work, we can cite Carrasco et al. [2011] which used a multiagent system [Wooldridge, 2002] with frames and ontologies to capture evidence of illegal trafficking of wild animals on social networks. In this study, they used a domain ontology and a frame related scene was modeled. However, they did not use a frame ontology.

We can also cite the work of Rodrigues et al. [2012] which did the same, but to conduct monitoring of graduates. The same author also made his model with agents but used Bayesian networks [Friedman et al., 1997] to assess the likelihood of a sentence to be a case of a certain semantic frame. Moreira and Salomão [2013] used an ontology to detect semantic frames. Scheffczyk et al. [2006a] presented a way of linking large lexical resources with world knowledge via ontologies. In [Scheffczyk et al., 2006b], the same author proposed a set of rules for mapping Frame Elements with elements of the SUMO[8] Ontology.

[8] protege.stanford.edu/ontologies/sumoOntology/sumo155.zip

3.2.7 Conclusions

This article showed how ontologies and frames can be used together for natural language processing. As an example, we used a job offer frame. The frame itself was mapped to the ontology and the transition to a domain ontology was done by means of relations. Through testing it was shown that the analysis of natural language can be made and important information of a given context can be stored in an ontology to make future analyzes. For this method to work it is necessary that the ontology is populated with elements that represent its meaning in the context of the problem.

As future work, the same method can be applied to other frames and other fields. In addition, the relationships frame-to-frame can be further explored in order to infer additional information. Can be used also other Frames, interrelated, for a wider range of information not covered by the Job Offer Frame.

Ackmowledgment

This work is financed by funding agencies FAPEMIG, CNPq and FUNARBE and by the Gapso Company.

3.3 Artigo III: Acompanhamento da Evolução Profissional de Egressos com Sistemas Multiagentes

Diego Fialho Rodrigues, Alcione de Paiva Oliveira,
Jugurta Lisboa Filho e Alexandra Moreira

Submetido à Revista para o Processamento Automático das Línguas Ibéricas (Linguamática)

Resumo

Várias instituições precisam coletar e armazenar informações sobre pessoas relacionadas com suas atividades. Dentre essas instituições destacam-se as instituições de ensino, que necessitam acompanhar a evolução profissional de seus egressos. Um sistema automatizado para realizar tal tarefa precisa superar vários desafios: coletar informações em bases de textos em linguagem natural, efetuar a extração de informação por meio de técnicas de processamento de linguagem natural, identificar entidades e os papéis exercidos por essas entidades na situação descrita, integrar informações heterogêneas e armazenar essas informações em uma base de fácil manipulação. Para realizar esta tarefa complexa que envolve diversas etapas, os sistemas multiagentes aparecem como uma alternativa para de solução escalável e modular. Este artigo descreve um modelo multiagente para obtenção de informação profissional sobre egressos. O modelo foi implementado e testado em uma base de mensagens trocadas em uma lista de e-mails de egressos. Os testes mostraram a viabilidade desta abordagem na solução de problemas de extração de informação em linguagem natural.

Abstract

Several institutions need to collect and store information about people related to their activities. Among such institutions there are the educational institutions that need to monitor the professional development of its graduates. An automated system for performing this task must overcome several challenges: collect information in natural

language databases, extract information through natural language processing techniques, identify the entities and the roles played by these entities in the situation described, integrate heterogeneous information and store this information on an easy handling database. Multiagent systems appear as a scalable and modular solution to accomplish this complex task that involves several steps. This paper describes a multiagent model to obtain professional information of graduates. The model was implemented and tested in a database of messages exchanged in a list of e-mails. The tests showed the feasibility of this approach in solving problems of information extraction in natural language.

3.3.1 Introdução

Várias instituições precisam coletar e armazenar informações sobre pessoas relacionadas com suas atividades. Dentre essas instituições destacam-se as instituições de ensino, que necessitam acompanhar a evolução profissional de seus egressos. Essas informações podem ser encontradas em redes sociais, listas de e-mail, blogs, sites de currículos disponibilizados publicamente, etc. Apesar da abundância de dados, existe a dificuldade de agregar todas estas informações, pois são disponibilizadas em diversos formatos, sem seguir nenhum padrão específico. Além disso, coletar essas informações de forma manual se mostra uma tarefa inviável pelo grande número de fontes de informações. Um sistema automatizado capaz de coletar e armazenar tais informações seria de grande utilidade. Porém, existem vários desafios a serem superados: coletar dados em bases totalmente textuais e não estruturadas, realizar inferências textuais, identificar indivíduos, integrar informações distribuídas, etc. Sistemas multiagentes (SMA) [Wooldridge, 2002] se apresentam como uma alternativa para resolver este problema de forma gradual e escalável. SMAs propiciam que sistemas possam operar de forma descentralizada, onde componentes podem ser adicionados de forma natural, com impactos mínimos ao sistema global, devido ao baixo acoplamento das unidades.

Este trabalho descreve um sistema capaz de monitorar informações sobre o local de trabalho de pessoas de acordo com as informações disponibilizadas na Internet em sites públicos. Mais especificamente, a meta é monitorar a evolução profissional de

egressos. No Brasil, o acompanhamento da trajetória profissional dos alunos, assim como sua evolução acadêmica é um dos critérios para avaliação dos cursos de graduação. Este artigo se baseia no modelo descrito em Rodrigues et al. [2012]. O modelo foi implementado na forma de um sistema multiagente e foram realizados testes, cujos resultados são apresentados neste artigo (seção 3.3.10). A construção do sistema segue vários passos, descritos nas seções que seguem:

- A criação de um modelo baseado em agentes com o propósito de permitir uma maior flexibilidade à solução (seção 3.3.4), uma vez que propicia o acréscimo de agentes mineradores de informação especializados em fontes particulares de informação.

- Criação de tabelas para armazenamento das informações coletadas em nuvens (seçã 3.3.5).

- Criação de um Web Service para a obtenção e manutenção da lista atualizada de egressos (seção 3.3.6).

- Elaboração de questionários para coletar informações adicionais dos alunos. O envio e processamento das respostas é feito de forma automática (seção 3.3.7).

- Para realizar a leitura e processamento de e-mails, foi necessária a criação de uma estrutura para leitura e normalização das mensagens com a utilização de diversos agentes (seção 3.3.8).

- Um frame semântico foi criado para ajudar no processamento das mensagens (que estão em formato de linguagem natural) (seção 3.3.8).

- Uma ontologia [Chandrasekaran et al., 1999] foi criada para modelar os elementos do domínio assim como suas relações (seção 3.3.8). Também foi gerada uma ontologia para modelar os frames semânticos (seção 3.3.8).

- Como critério de seleção das mensagens foi utilizado uma rede Bayesiana para estabelecer a probabilidade de uma sentença pertencer à cena descrita pelo frame

semântico. (seção 3.3.8).

- Um estudo de caso foi realizado (seção 3.3.9) e testes foram realizados para classificar as evidências encontradas (seção 3.3.10).

Na próxima seção é abordado o problema de acompanhamento de pessoas e o que tem sido realizado para solucioná-lo.

3.3.2 Acompanhamento de pessoas e trabalhos correlatos

A busca e coleta de informações sobre pessoas tem se tornado uma tarefa essencial para as empresas. A tarefa de procurar especialistas em determinada área do conhecimento foi introduzida em *The Fourteenth Text Retrieval Conference* - TREC 2005 [Craswell and de Vries, 2006]. A busca por pessoas também foi uma dos assuntos do *the Fourth International Workshop on Semantic Evaluations* [SemEval, 2007], organizado pela *Association for Computational Linguistics* (ACL). Artiles et al. [2007] apresentou um critério para avalizar a performance de sistemas que buscam pessoas na Internet. Elmacioglu et al. [2007] apresentou um sistema que elimina ambiguidade em pesquisas relacionadas à nome de pessoas na Web através de agrupamento. Popescu and Magnini [2009] propôs meios de amenizar o problema de se estabelecer referências errôneas nas pesquisar por pessoas. Balog et al. [Balog et al., 2009; Balog and de Rijke, 2008] também atacou o problema de resolução de nome de pessoas usando técnicas de agrupamento. A maioria dos estudos tratando da associação de pessoas à documentos usam técnicas de agrupamento para eliminar ambiguidades. A suposição básica desta técnica é que documentos similares tendem a representar a mesma pessoa. Alguns dos principais métodos de agrupamento são [Balog et al., 2009]: Single Pass, k-Means, agglomerative clustering e probabilistic latent semantic analysis (PLSA). Os primeiros três métodos possibilitam variações do método de agrupamento tradicional (difere em termos de eficiência e qualidade). Esses métodos também se baseiam no fato de que diferentes documentos possuem os mesmos termos associados ao mesmo indivíduo. Carrasco et al. [2011] apresentou um sistema para identificar o tráfico de animais silvestre em redes

sociais a partir da análise das mensagens trocadas pelos usuários. Apesar de não monitorar pessoas, esse trabalho também utilizou frames e ontologias para identificação do contexto do enunciado e do significado dos itens lexicais. Moreira [2012] usou ontologias para identificar enunciados relacionados ao Frame VIAGEM. Scheffczyk et al. [2006a] apresentou uma maneira de ligar grandes recursos lexicais com o conhecimento de mundo através de ontologias. Em [Scheffczyk et al., 2006b], o mesmo autor propôs um conjunto de regras para mapear FEs com os elementos da ontologia SUMO[9].

O monitoramento de egressos pode ser considerada como uma tarefa relacionada ao acompanhamento de pessoas. O sucesso profissional de ex-alunos de instituições de ensino pode ser atribuído em parte ao treinamento recebido nessas instituições. A prática comumente utilizada pelas instituições é o envio de mensagens solicitando que os egressos atualizem, voluntariamente, seus dados. No entanto, essa é uma prática que não permite um acompanhamento eficiente, uma vez que parte da premissa que o banco de endereços (e-mail ou endereço físico) já devem estar atualizados e de que os egressos estarão dispostos a dedicar parte de seu tempo para responder e enviar formulários. Por isso, um sistema automatizado que busque informações sobre egressos em várias fontes seria de grande utilidade. Obviamente que, por questões éticas e de privacidade, este tipo de sistema somente deve buscar informações em repositórios públicos, disponíveis diretamente e com o consentimento das partes envolvidas. Para implementar um sistema dessa natureza é necessário utilizar técnicas de processamento de linguagem natural e de processamento distribuído, notadamente de sistemas orientado a agentes, com o intuito de tornar o sistema mais modular e escalável.

3.3.3 A abordagem Multiagente

Sistema Multiagente é uma área dentro da Ciência da Computação que lida com aspectos de Sistemas Distribuídos aplicados à Inteligência Artificial [Bordini et al., 2001]. Esta área surgiu de observações feitas em alguns sistemas naturais, tais como colônias de formigas, onde se observa um comportamento, aparentemente inteligente, emergir à

[9]http://protege.stanford.edu/ontologies/sumoOntology

partir da interação dos seus elementos [Bolzan and Giraffa, 2002; Hübner et al., 2004]. A pesquisa sobre SMA foge dos paradigmas tradicionais, pois tem como objeto de estudo a coletividade e não o individual. No campo de Sistemas Multiagentes, estuda-se o comportamento de um grupo organizado de agentes autônomos, que cooperam para resolver problemas que estão além das capacidades individuais de resolução. Um dos requisitos da solução buscada para o problema de acompanhamento de egressos é que ele possa analisar informações oriundas de diversas fontes. Apesar de utlizamos apenas três fontes (web service da Universidade, lista de e-mails e respostas de questionários), os sistema deve ser preparado para a inserção de novas fontes. Por isso, os analisadores apropriados para extrair as informações dessas fontes devem ser adicionados ao sistema de forma transparente e com o mínimo de impacto sobre os outros módulos. Para atender esse requisito, a utilização de um sistema com arquitetura baseada em agentes parece apropriada. Sendo assim, é possível alterar um agente especializado em extrair informações de uma determinada base ou substituir uma determinada técnica de processamento de linguagem natural sem interferir em outro agente do sistema. Além disso, uma plataforma multiagente oferece a possibilidade de incluir novos agentes, tornando o sistema naturalmente escalável. Na arquitetura proposta os dados são enviados para um agente que armazena e gerencia as informações em um repositório central. Existe também a possibilidade do sistema se beneficiar do uso de um hardware dotado de processamento paralelo, onde cada agente poderia ser executado em um processador distinto.

3.3.4 O Modelo

O sistema proposto tem a função de abrigar diversos agentes que serão inseridos gradativamente. Cada agente será capaz de buscar informações em uma fonte distinta e de forma assíncrona, sendo que as informações serão armazenadas em um repositório único. Desta forma a arquitetura multiagente se apresenta como uma solução natural. O sistema não é composto apenas por agentes buscadores de informação. Existem vários tipos de agentes, dos mais simples (como fazer comunicação com um *web Service*) aos

mais complexos (responsável pelo processamento de linguagem natural). O objetivo é fazer o sistema ser o mais modular possível, de forma que a inserção ou remoção de um agente não comprometa o funcionamento do sistema como um todo. Isto se deve pelo fato da interação entre os agentes ser feita por meio de troca de mensagens e, portanto, permitindo um menor acoplamento entre os agentes.

O modelo possui seis agentes, a saber: (i) *University Web Server*, (ii) *Email Reader*, (iii) *Questionnaire Sender*, (iv) *Questionnaire Answers Reader*, (v) *Database Manager* e (vi) *NL Processor*. Cada um destes agentes pode comunicar com agentes secundários para auxiliar na execução de suas tarefas.

O agente *University Web Server* é responsável por buscar na base de dados da instituição de ensino as informações sobre os alunos que acabaram de se formar. A consulta é feita a cada seis meses, que é o intervalo entre cada formatura. A lista de graduados obtida pelo agente serve de ponto de partida para todas as outras buscas. A lista contém informações tais como nome e e-mail que servirão para a identificação dos egressos nos resultados das buscas realizadas pelos outros agentes.

O agente *Email Reader* tem a função de ler as mensagens de e-mail contidas na listas de discussão do grupos de ex-alunos. Ele foi projetado para buscar por informações sobre a evolução profissional dos egressos. Uma análise da base de mensagens revelou que é comum que os egressos informem suas atividades atuais, tais como, empresa onde trabalham e cidade onde moram e essa informação forneceu a base para a definição do esquema que seria utilizado para a extração dos dados. O agente *Email Reader* possui dois agentes auxiliares. O primeiro, tem a função de colocar os diversos formatos de e-mail em uma única forma canônica. O segundo agente, tem a função de extrair as diversas sentenças contidas numa única mensagem de email.

O agente *Questionarie Sender* envia um questionário para o email de cada egresso. Os ex-alunos preenchem as informações e reenviam o formulário. O *Questionarie Answers Reader*, por sua vez, tem a função de processar as respostas e atualizar a base de dados.

O agente *NL Processor* processa todas as sentenças fornecidas pelo agente *Mail*

Reader. Este agente usa técnicas de processamento de linguagem natural para tentar identificar evidências de vínculo entre um egresso e uma instituição.

O agente *Database Manager* coordena o repositório de dados. Nenhum dos outros agentes pode atualizar da base de dados diretamente. Todas as atualizações são feitas por intermédio do agente *Database Manager*.

A Figura 3.10 mostra a dependência entre os agentes. Foi utilizado o diagrama de dependência de agentes da metodologia Tropos [Giunchiglia et al., 2002]. Os agentes principais e secundários serão detalhados na seções que seguem.

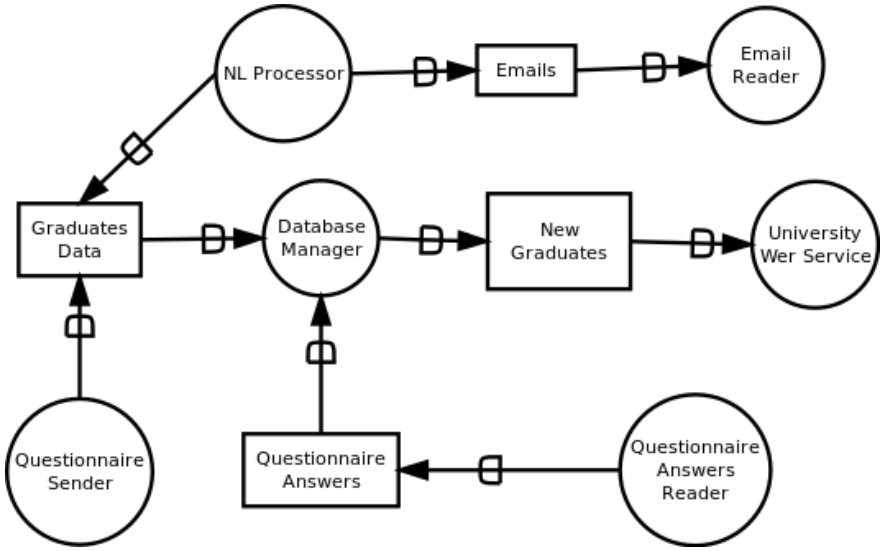

Figura 3.10: Dependência entre os agentes.

3.3.5 Armazenamento em Nuvens

Como dito anteriormente, o Agente Database Manager coordena todas as operações ao repositório de dados. Desta forma, pode-se trocar a tecnologia de armazenamento sem perpetuar esta mudança para os demais agentes. O armazenamento das informações foi feito em nuvens [Armbrust et al., 2010]. Assim, podemos delegar a tarefa de armazenamento para terceiros, construindo apenas uma interface de acesso aos dados. Os agentes que serão descritos nas próximas seções serão responsáveis por buscar informações de egressos na Internet e inseri-las neste repositório. O repositório também servirá de base

para consultas, como buscas por nome ou e-mail. Foi utilizado o Google Drive[10] e as informações foram dividas em 3 tabelas como mostra a figura 3.11.

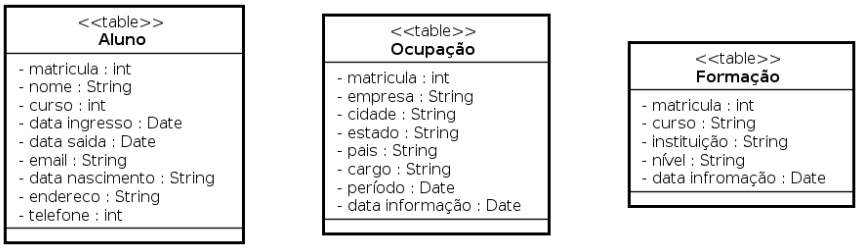

Figura 3.11: Tabelas armazenadas em nuvem.

A primeira tabela, Aluno, possui as informações básicas sobre os egressos, como matrícula, nome, e-mail, entre outros. Esta tabela é alimentada pelo Web Service da Universidade que será descrito na próxima seção. A tabela de ocupação armazena o históorico de empregos que o egresso ocupou e é vinculado à tabela Aluno através do campo matrícula. A tabela Formação guarda as informações referentes ao cursos realizados, como mestrado, MBA, doutoraro, etc. Esta tabela também é vinculada à tabela aluno por meio do campo matrícula.

3.3.6 Web Service da Universidade

O agente *University Web Service* é o único que conta com um agente externo à plataforma multiagente. Ele fica hospedado no servidor de banco de dados da instituição de ensino. A cada seis meses, novos alunos se formam e vão para o mercado de trabalho e essas informações são atualizadas na base de dados. A função do agente é capturar e enviar a lista dos novos graduados ao Sistema Multiagente, o que servirá de base para os demais agentes.

O *web service* é bem simples. Basta o cliente enviar uma requisição, informando um usuário e senha, que a resposta trará informações como matrícula, nome, e-mail, data de entrada e saída da universidade.

Além do agente *web Service*, externo ao sistema, existe um agente que faz a interação com o serviço e o agente interage com o agente *Database Manager* para gravar as novas

[10]https://drive.google.com

informações no repositório de dados.

3.3.7 Questionários

Dentre todas as formas de se obter informações sobre as pessoas, talvez a mais simples e direta seja através de questionários. E é isso que o agente *Questionaire Sender* faz de tempos em tempos, enviando um e-mail com um questionário para a lista de ex-alunos do curso de Ciência da Computação da instituição de ensino. Este e-mail contém uma mensagem explicando a motivação do questionário e um link para um formulário com todas as perguntas do questionário. O formulário contém perguntas tais como:

- Qual é o seu nome completo?
- Qual é o seu email?
- Informe algum curso que você tenha feito ultimamente.
- Informe o nome da instituição onde o curso foi feito.
- Informe o tipo do curso (MBA, Lato Sensu, Mestrado, Doutorado, Outro).
- Informe o nome de uma empresa que você trabalha ou tenha trabalhado.
- Qual é o cargo que você ocupa?
- Em que país fica localizada a empresa?
- Em que estado fica localizada a empresa?
- Em que cidade fica localizada a empresa?
- Quando você entrou nesta empresa?
- Quando você saiu da empresa?

Além destas informações, também é armazenada a data em que o formulário foi submetido. O egresso responde o questionário e as informações são enviadas automaticamente para uma tabela no repositório de dados. Contudo, as informações precisam ser

preparadas para serem inseridas no repositório. Para realizar essa tarefa entra em cena o agente *Questionnaire Answers Reader*. Este agente tem a função de ler a planilha com as respostas e atualizar os dados do graduado. O agente age em intervalos de tempos, por exemplo, de mês em mês.

Os questionários são apenas uma forma de coletar mais informações sobre os egressos e supõe-se que são informações confiáveis. No entanto, nem todas as pessoas se dão ao trabalho de responder questionários. Sendo assim, se tornam necessárias outras formas de coletar informações que serão mostradas nas seções que seguem.

3.3.8 Processamento de Linguagem Natural

Uma outra fonte de informação sobre os egressos são as mensagens de e-mails. Analisando a lista e e-mails dos ex-alunos, percebe-se que, muitas vezes, as pessoas deixam informações sobre seus empregos nestas mensagens. Um exemplo comum são as ofertas de emprego. Geralmente, uma pessoa divulga uma vaga na empresa onde trabalha postando uma mensagem para a lista e e-mail. Diferentemente das informações adquiridas com o *Web Service* e com os questionários que estão em um formato estruturado, as informações contidas nas mensagens de e-mails estão em linguagem natural. Por isso, é preciso realizar um processamento com uso de técnicas de PLN para ser possível extrair informações relevantes. Estas técnicas serão descritas mais adiante neste artigo. Porém, antes de processar as mensagens foi necessário criar mecanismos para extrair as mensagens de e-mails das lista. Uma vez que existem vários formatos de e-mails e vários protocolos de comunicação foi necessário separar esta etapa da etapa que realiza o processamento das sentenças contidas nas mensagens.

Leitura de Emails

Para realizar a leitura dos e-mails foram utilizados três agentes. A Figura 3.12 mostra o fluxo da informações entre os agentes. O primeiro agente é o *Mail Reader*, ele é responsável por fazer a conexão com o servidor de e-mail e buscar as mensagens em seus formatos originais. As mensagens são lidas e armazenadas em um buffer de e-

mails. São capturadas informações como remetente, destinatário, data de envio, data de recebimento, assunto e corpo da mensagem. Esse agente se conecta com o servidor de e-mails em certos intervalos em busca de novas mensagens.

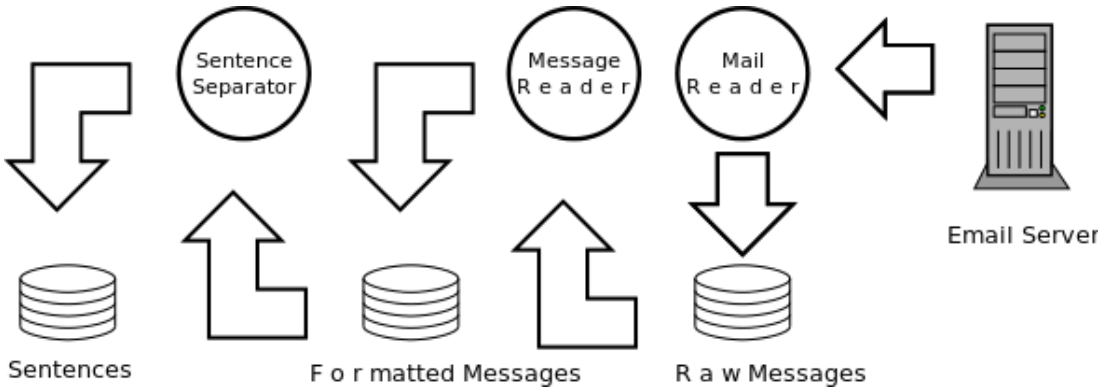

Figura 3.12: Mensagens sendo processadas.

Existem vários formatos de mensagem de email. Existem mensagens em texto puro, em formato html, com imagens, em formato de array de mensagens (chamado de multipart), etc. Para lidar com esta diversidade de formatos foi criado o agente *Message Reader*. Este agente lê as mensagens do buffer de e-mails (alimentada pelo agente *Mail Reader*) e verifica o tipo do formato da mensagem. Este agente tem ao seu dispor vários leitores para os diversos formatos. Ele, então, escolhe o leitor adequado para o formato e processa a mensagem, traduzindo as mensagens para um formato canônico. Estas mensagens são colocadas em um segundo buffer, para serem processadas pelo próximo agente. Alguns formatos que não interessam ao sistema, tais como imagens, são simplesmente descartados.

O terceiro agente, *Sentence Separator*, tem a função de quebrar as mensagens em sentenças. Isto é feito através da busca de caracteres especias como '.', '?', '!', etc. Existem alguns problemas na separação de sentenças. Por exemplo, as abreviações, como 'Sr.', podem ser erroneamente identificadas como um separador de sentenças. Além disso, os e-mails, não são sempre escritos com formalidade, as pessoas se esquecem de colocar ponto final com muita frequência. Para resolver o primeiro problema, foi criada uma lista de abreviações, que são consultadas toda vez que um ponto é encontrado no texto.

Sobre o segundo problema, fica difícil tentar inferir se uma quebra de linha, mesmo sem ponto final, pode ser considerado o fim de uma sentença, o que pode acarretar em perda de evidências.

As informações são armazenadas em um terceiro buffer que servirá de entrada para os processadores de linguagem natural. É importante destacar que todos esses três agentes são executados paralelamente, no estilo pipeline. Assim que o agente *Mail Reader* insere alguma informação no buffer, o leitor de mensagens já pode começar a processar. O mesmo vale para o *Sentence Parser*. O sistema permite que se leia e-mails de mais um servidor, apenas, bastando para isso, instanciar outro leitor de e-mails, sem causar maiores impactos no sistema.

Frames Semânticos

Verificando a lista de e-mails, pode-se perceber vários padrões ou esquemas de comunicação. Alguns destes padrões evidenciam a ligação de uma pessoa com uma certa organização. Por exemplo, foi notado que é comum as pessoas divulgarem vagas de emprego em sua empresa ou que estão fazendo algum curso de pós-graduação em alguma instituição de ensino. De certa forma, os componentes são quase sempre os mesmos: o egresso, a empresa ou instituição de ensino, a cidade, a data. O objetivo é identificar estes padrões entre as mensagens para capturar as informações relevantes ao problema.

Para melhor identificar estes padrões, foram utilizados os Frames Semânticos [Fillmore, 2006]. De acordo com Baker et al. [1998], Frames Semânticos são estruturas conceituais que descrevem um tipo particular de situação, objeto ou evento, juntamente com seus participantes. Estes participantes são chamados Elementos de Frames (*Frames Elements* ou FEs). Os FEs são divididos em dois grupos: os nucleares (*core*) e não-nucleares (*non-core*). Os elementos nucleares sempre ocorrem, implícita ou explicitamente. Os elementos não-nucleares podem ocorrer ou não. Além disso, cada Frame é evocado por uma Unidade Lexical (*Lexical Unit* ou LU), que é uma palavra com um significado (par forma-significado). Os FEs agem como argumentos para as LU, completando o sentido apresentado pela palavra.

Uma vez que os frames são definidos, pode-se utilizá-los para reconhecer sentenças relacionados aos frames pela identificação de seus elementos. No caso deste trabalho, estas informações são as relacionadas à progressão profissional do egresso. Portanto, os FEs são as informações que serão analisadas e inseridas no repositório de dados.

A criação dos Frames foi feita através da análise de um *corpus* construído a partir da mensagens contidas na lista de discussão de ex-alunos. Um *corpus* pode ser entendido como um conjunto de textos escritos de uma determinada língua e que serve com base de análise [Tognini-Bonelli, 2001]. Foram lidas cerca de 5 mil mensagens enviadas entre 1999 e 2011. A análise da lista de e-mails revelou que é comum para um egresso oferecer vagas de trabalho para a companhia onde ele trabalha. Junto à oferta, o ex-aluno pode mencionar outros itens importantes, como a cidade onde a empresa se localiza, a data que ele entrou na empresa, a função exercida, etc. Geralmente a sentença está na primeira pessoa e o nome da companhia é mencionado. Estas observações foram usadas para construção de um frame, denominado de JOB OFFER. O Frame está definido na Figura 3.13 junto com alguns exemplos. Foi seguida a mesma notação utilizada no site do projeto FrameNet[11] que se propõe a construir uma base lexical baseado na semântica de frames.

No sistema proposto, cada frame será atribuído a um agente distinto. No momento, apenas o frame JOB_OFFER foi criado, mas, futuramente, outros frames serão desenvolvidos. O agente correspondente a cada frame fará a busca pelas sentenças no buffer (alimentada pelo agente *Sentence Separator*). O processo consiste em buscar sentenças que contenham alguma unidade lexical do respectivo frame. Para um ser humano, ler uma sentença e identificar os elementos do frame e unidades lexicais não parece ser uma tarefa muito difícil. No entanto, não é tarefa simples para um computador. Os detalhes de como um frame é identificado e como os elementos são encontrados e classificados será detalhado nas subseções que seguem.

[11] https://framenet.icsi.berkeley.edu/fndrupal/

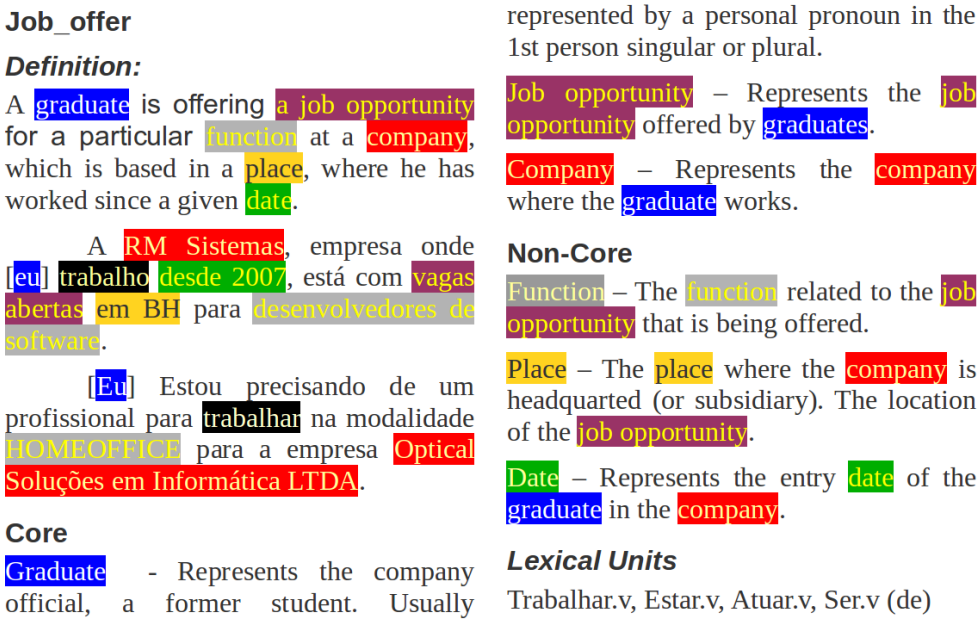

Figura 3.13: O Frame Job Offer.

A Ontologia de Domínio

Frames podem ser usados para fazer a análise semântica dos elementos mas eles não necessariamente definem a natureza ou tipo desses elementos. Por exemplo, o trecho de texto "em São Paulo" faz referência a um lugar, contudo não é possível saber se é uma cidade, um estado ou a que país pertence. Para classificar os elementos do domínio, foi usada uma ontologia de domínio [Uschold and Gruninger, 1996]. Uma ontologia define os conceitos relevantes ao problema, assim como todas as relações entre estes conceitos. Além disso, uma vez que ontologias permitem expressar diversos tipos de relações entre os conceitos do domínio, é possível usar essas relações para a realização de inferências sobre os indivíduos do domínio, como por exemplo, pesquisar todas as empresas que estão localizadas em uma certa cidade.

A ontologia de domínio que desenvolvemos tem o propósito de descrever os elementos que ocorrem no contexto das relações empregatícias entre egressos e empresas. Esta ontologia possui elementos representando instituições onde pessoas podem ser empregadas, lugares (países, estados e cidades) e funções de trabalho, etc. Também foi definida uma estrutura de tempo para representar intervalos, mais especificamente para

representar o tempo de duração de um vínculo entre uma instituição e uma pessoa.

Para ajudar a classificar melhor os elementos, a ontologia foi construída sobre outras duas ontologias mais gerais (ontologias topo): a ontologia *DUL* (Dolce Ultralight)[12] e a ontologia *Time*[13]. Em função disso, não houve necessidade de se criar muitas classes (conceitos) e propriedades (relações entre os conecitos). A maioria das classes usou propriedades já definidas nas outras duas ontologias, como na relação *is part of* entre *Department* e *Organization*. A ontologia desenvolvida é mostrada na Figura 3.14. As partes (a), (b) e (c) contém as classes enquanto que as partes (d) e (e) exibem as propriedades. Os elementos criados por esta pesquisa estão em negrito.

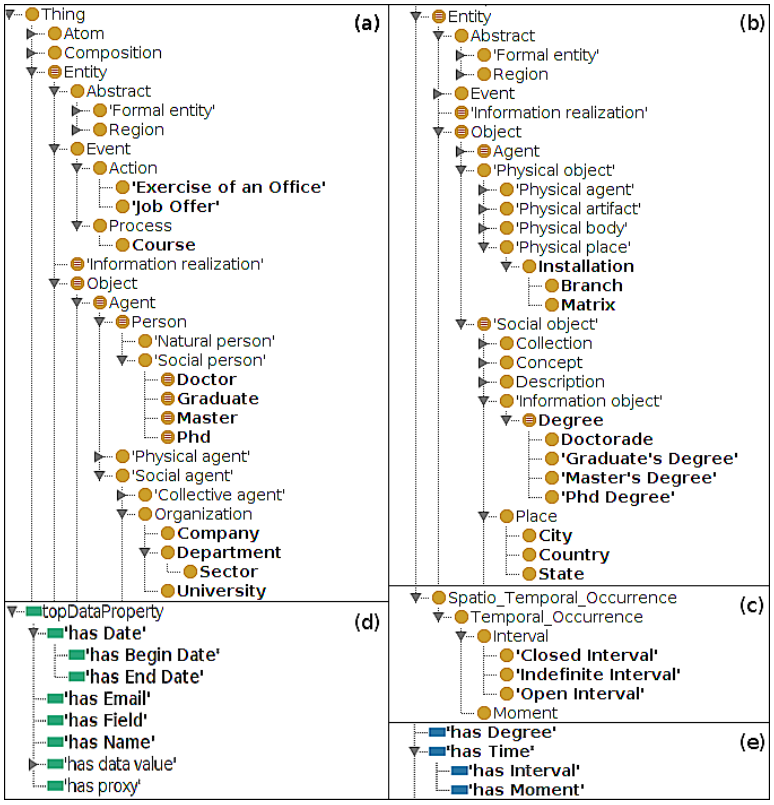

Figura 3.14: A ontologia de domínio construída sobre as ontologias DUL e Time. A, B e C: as classes. D e E: as propriedades.

[12] www.loa.istc.cnr.it/ontologies/DUL.owl
[13] raw.github.com/RinkeHoekstra/lkif-core/master/time.owl

A Ontologia de Frames Semânticos

Os frames e as ontologias formam os dois pilares fundamentais desta proposta. Os frames são usados para definir a cena ou contexto da ocorrência do enunciado, estabelecendo desse modo semântica dos elementos do enunciado. A ontologia de domínio, define o tipo do conceito subjacente ao enunciado, o que possibilita a formalização e inferências sobre os elementos de estudo. No entanto, ainda falta um elemento que ligue esses dois aspectos. Esse elemento é a chamada Ontologia de Frames Semânticos.

Neste trabalho o ponto de partida do processamento da linguagem natural é a análise dos frames. Ou seja, verificar se uma sentença denota a ocorrência de um frame. Para um ser humano, esta é uma tarefa relativamente fácil, desde que ele tenha conhecimento suficiente do contexto. Se uma pessoa ler uma sentença que contenha uma oferta de emprego, é bem provável que reconheça que parte se refere ao local de trabalho, para que cargo é a vaga, qual é a cidade, etc. Mas não é isso que acontece quando um computador processa um texto. Para que este reconhecimento seja feito, deve haver uma ligação entre os trechos do texto e algum objeto em nossa ontologia de domínio.

Para resolver este problema, a ontologia de frames foi utilizada para especificar todos os conceitos inerentes aos frames semânticos, tais como unidades lexicais, elementos de frames e o próprio frame. A figura 3.15 mostra uma parte simplificada da ontologia de frames semânticos. A classe *Syntax* representa os trechos de texto. Esta classe possui as relações *evokes* e *fillerOf* com as classes *Frame* e *FE*, respectivamente. Um frame possui vários FEs, representados pela propriedade *hasFE*. Além da relação de herança, naturalmente representada na ontologia, existem outras relações entre os frames, a saber: (i) *Perspective on*, (ii) *Subframe*, (iii) *Precedes*, (iv) *Inchoative of*, (v) *Causative of*, (vi) *Using* e (vii) *See also*. Esta parte da ontologia foi baseada no trabalho de Scheffczyk et al. [2006a]. Também é importante ressaltar que, apesar de considerarmos uma ontologia à parte, ela foi incorporada junto à ontologia de domíno, extendento também a ontologia DUL. Mais especificamente, todas as classes ficaram como filhas da classe *Abstract* da ontologia DUL.

A propriedade *evokes* tem como domínio um segmento de texto, denominado de

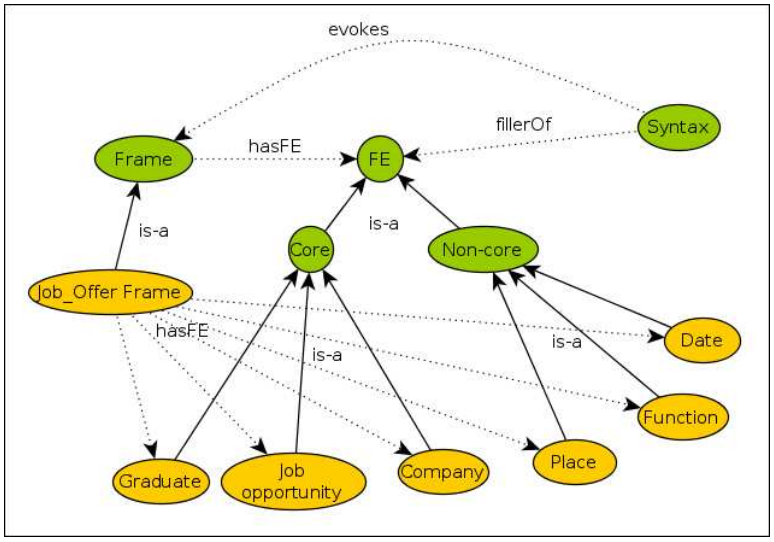

Figura 3.15: A Ontologia de Frames Semânticos (Simplificada).

Span e como imagem, uma instância do tipo *Frame*. O *Span* será justamente uma das unidades lexicais do frame. Uma vez encontrada a unidade lexical, pode-se verificar se aquela sentença é ou não um frame realizando a busca de seus FEs. Através da relação *hasFE* é possível saber quais são os FEs que devem ser procurados.

A ligação entre um elemento de frame e uma classe da ontologia de domínio é feita através relação *hasSemanticType* [Scheffczyk et al., 2006b]. Esta relação liga uma instância do tipo *Span*, que representa um trecho texto, a seu significado no contexto do problema, que, por sua vez, é uma classe da ontologia de domínio ou, em certos casos, uma instância da classe. Por exemplo, o trecho *Rio de Janeiro* possui uma relação *hasSemanticType* com a instância *city.rio_de_janeiro*, que pertence à classe *City*. Desta forma, o computador poderá inferir que aquela parte do texto se refere à uma cidade. Por sua vez, a classe *City* é filha da classe *Place* e logo se chega a conclusão que aquele texto é um candidato a ser marcado como um FE do tipo *Place*. A figura 3.16 mostra a relação entre estas instâncias.

Redes Bayesianas

Foi mostrado um modo de anotar as sentenças com os FEs e LUs. Contudo, é preciso detectar a cena de fundo (frame) relacionada à sentença. É o frame que atribui o

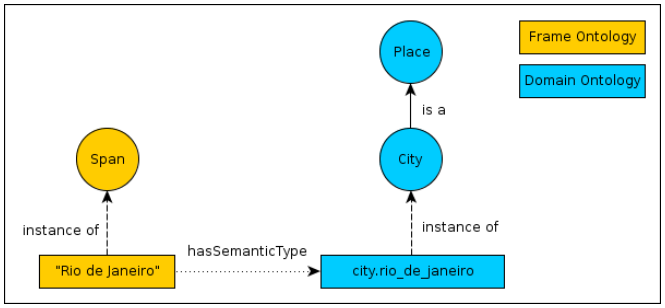

Figura 3.16: A propriedade “hasSemanticType”.

significado final à sentença [Fillmore, 2006]. Portanto, é preciso definir uma técnica para realizar o mapeamento entre a sentença e seu frame de fundo. Moreira [2012] propôs o uso de redes Bayesianas para realizar esse mapeamento. Como já mencionado, Balog and de Rijke [2008] utilizaram o teorema de Bayes para determinar a probabilidade de uma pessoa estar associada a um certo documento. Porém, este trabalho foca no casamento no nível de sentenças e não no nível de documentos.

Um frame semântico possui vários componentes, alguns obrigatórios e outros opcionais. Sendo assim, um frame pode ser identificado por meio da presença de suas unidades lexicais e elementos de frame. Neste processo de identificação, alguns elementos possuem mais peso, como é o caso das unidades lexicais e elementos de frame nucleares. Outros possuem um peso menor, como os elementos de frame não nucleares.

No estudo, foram utilizadas Redes Bayesianas para classificar as sentenças conforme proposto em Moreira [2012]. De acordo com Friedman et al. [1997], Redes Bayesianas são grafos acíclicos e direcionados que codificam a distribuição de probabilidade conjunta sobre um conjunto de variáveis aleatórias. Cada variável é representada por um vértice. A correlação entre as variáveis é representada por arcos. Para cada variável, existe uma tabela de probabilidade representando a distribuição local de probabilidades dados seu pais (vértices que dão origem aos arcos entrantes). O modelo foi construído com base no frame semântico da seção 3.3.8.

Frames são evocados por unidades lexicais [Ruppenhofer et al., 2010]. A unidade lexical, por sua vez, possui uma estrutura argumental, sendo que os elementos de frames se comportam como parâmetros da LU. Com base nestas afirmações, foi montada

uma Rede Bayesiana sobre o frame de oferta de trabalho, sendo que a ocorrência dos elementos de frames e da unidade lexical na sentença funcionam como evidencias para a ocorrência do frame. Sendo assim, os FEs e a LU são as variáveis independentes e o frame, a variável dependente. A rede foi construída usando o software Weka [Hall et al., 2009]. Uma lista de 150 sentenças foi anotada e, então, classificada como uma ocorrência ou não do frame. De todos os algorítimos disponíveis no aplicativo, o *Tree Augmented Naive Bayes* (TAN) [Friedman et al., 1997] apresentou o melhor resultado (94.0% dos casos corretamente classificados). A rede é ilustrada na figura 3.17. Os vértices representam cada um dos FEs, a LU e o próprio frame.

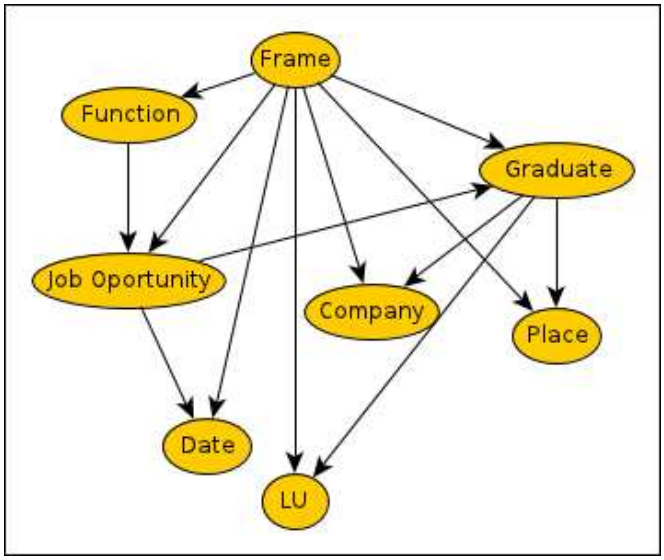

Figura 3.17: A Rede Bayesiana sobre o Frame Job_Offer.

O comportamento do modelo é sempre procurar por sentenças que contenham LUs do frame em questão. Uma vez que a unidade lexical é encontrada, a tarefa passa a ser encontrar elementos de frame. Então, a sentença é classificada de acordo com a presença ou ausência de cada um dos elementos do frame. Quanto mais elementos são encontrados, maior é a probabilidade de a sentença ser classificada como exemplo de um frame.

3.3.9 Estudo de Caso

Nesta seção, será mostrado um exemplo de como uma sentença é processada. Primeiramente, é feita a análise no nível de frames semânticos. Todos os elementos do frame são identificados. Em seguida, estes elementos são ligados com objetos ou classes da ontologia de domínio através da relação *hasSemanticType*. Em alguns casos, a relação tem como imagem uma instância e, em outros casos, uma classe. Por exemplo, os FEs do tipo *Place* irão possuir uma relação com as instâncias da classe *City*. É claro, que um lugar tem um significado muito mais genérico que uma cidade. No entanto, neste caso em particular, o importante é estabelecer um vínculo com cidades. Além disso, é importante saber a qual estado e país pertence esta cidade. Isto é possível descobrir através da relações que uma instância da ontologia de domínio possui com outras instâncias.

Em outros casos, porém, não é preciso saber as características detalhadas sobre os objetos, como é o caso da função exercida. Nesse caso é preciso saber apenas que um certo segmento da sentença possui o significado de uma ocupação profissional. Por isso, a relação é feita com a classe.

A tabela 3.3 mostra as classes com as quais os FEs se ligam e se esta relação é com uma classe apenas ou com um instância.

FE	Tipo Semântico	Imagem
Job_Offer.Graduate	Person	Classe
Job_Offer.Job_Opportunity	Job Offer	Classe
Job_Offer.Company	Company	Instância
Job_Offer.Place	City	Instância
Job_Offer.Function	Role	Classe
Job_Offer.Date	Interval	Instância

Tabela 3.3: Elementos de Frame e seus Tipos Semânticos.

No exemplo será usada a seguinte frase (o nome da empresa foi omitida por questões de privacidade):

A X, empresa onde eu trabalho está com algumas vagas em aberto.

Aqui, considera-se que todo o processo de busca na lista de e-mails já foi feito. Obtém-se, assim, um conjunto de sentenças para serem submetidas ao analisador sintático. Cada

sentenças é lida e o primeiro passo é buscar por unidades lexicais. Existe uma lista de instâncias da classe *Span* que possuem uma relação do tipo *evokes* com o frame Job_Offer. Uma vez que alguma unidade lexical é encontrada, o texto é marcado e segue-se para a busca de cada FE.

O segundo passo é procurar por elementos de frame. Primeiramente, busca-se os elementos nucleares e, depois, os não-nucleares. Para fazer esta tarefa, deve-se procurar as classes que herdam da classe *Frame Element*. Mais especificamente, busca-se por subclasses de *Job Offer FE*. Cada uma dessas entidades possui uma relação *hasFE* com o frame em questão. Cada uma dessas classes possui a propriedade *hasSemanticType*. Esta relação conecta algum FE com alguma classe da ontologia de domínio.

Uma vez que se sabe o tipo do FE, é preciso procurar *Spans* cujo tipo semântico sejam do mesmo tipo. Desta forma, obtém-se todos os candidatos pra preencher o elemento de frame. Agora, tudo que é preciso fazer é verificar se algum destes *Spans* estão presentes na sentença. Um detalhe desta busca é que alguns elementos precisam ser precedidos por algumas palavras-chave. Por exemplo, elementos do tipo *Place* devem ser precedidos de palavras como *em*, *está localizada em*, etc.

Quando o casamento do FE é feita, uma instância do elemento de frame é criada. Uma relação to tipo *fillerOf* é feita entre o *Span* e o FE. No exemplo, após a identificação dos elementos, a sentença fica anotada da seguinte forma:

A [X $_{COMPANY}$*], empresa onde [eu* $_{GRADUATE}$*] [trabalho* $_{LU}$*] está [com algumas vagas em aberto* $_{JOB\ OPPORTUNITY}$*].*

Após a anotação da sentença é possível explorar as instâncias da ontologia de domínio. Os elementos textuais agora possuem um significado no contexto do problema. São identificadas as classes às quais eles pertencem e várias de suas propriedades. O elemento *X* possui uma relação com uma instância da classe *Company*. Este objeto, por sua vez, possui várias propriedades, como localização, proprietários, ramo de atuação, etc.

Depois que a sentença é anotada e os objetos da ontologia de domínio são iden-

tificados, é preciso decidir se esta sentença é mesmo um caso de oferta de trabalho. Como dito na seção 3.3.8, não existe uma forma categórica de se verificar se este é o caso, devido à polissemia inerente à linguagem natural. Podem existir vários indícios em função dos elementos presentes, porém, a sentença pode não estar relacionada com o frame em questão. Neste trabalho foi utilizada uma rede Bayesiana para tomar tal decisão, conforme sugerido por Moreira [2012]. Dada uma sequência de flags indicando ou não a presença dos elementos do frame, obtém-se a probabilidade condicional que indica as chances da sentença estar relacionada com o frame. Para este exemplo, os flags estão na tabela 3.4. Quando a rede foi treinada, foram considerados os como verdadeiros apenas os elementos que estavam explicitamente presentes.

Elemento	*Texto*	*Valor*
Lexical Unit	trabalho	Verdadeiro
Graduate	eu	Verdadeiro
Job Opportunity	com algumas vagas em aberto	Verdadeiro
Company	X	Verdadeiro
Function	-	Falso
Place	-	Falso
Date	-	Falso

Tabela 3.4: Presença dos FEs e LU.

Na sentença de exemplo, a probabilidade calculada pela rede Bayesiana foi de 97.67%. é importante notar que vários frames podem ser adicionados ao sistema e, cada, um deverá ter, sua própria rede Bayesiana.

3.3.10 Testes e Resultados

Para a realização dos testes foi criado um sistema multiagente descrito no modelo. Todos os agentes foram implementados, contudo os testes aqui apresentados abrangem apenas a parte de processamento de linguagem natural. O objetivo dos testes foi o de medir a capacidade do sistema de realizar a interpretação das sentenças em comparação com a avaliação de um ser humano. As avaliações tanto do sistema quanto as feitas manualmente geram dois resultados possíveis: positivo (é um caso) e negativo (não é um caso). Quando as duas avaliações são confrontadas, existem quatro resultados possíveis, listados na tabela 3.5. Estes resultados foram plotados em um gráfico ROC (*Receiver*

Sistema	*Humano*	*Avaliação*
Positivo	Positivo	Verdadeito Positivo
Positivo	Negativo	Falso Positivo
Negativo	Positivo	Falso Negativo
Negativo	Negativo	Verdadeiro Negativo

Tabela 3.5: Possíveis valores na comparação entre a avaliação do sistema e a avaliação humana.

Operating Characteristics) [Fawcett, 2006]. A curva ROC exibe a taxa de verdadeiros positivos (eixo y) e de falsos positivos (eixo x). Desta forma, quanto mais próximo é um resultado do ponto (0, 1), melhor a classificação. Seguindo o mesmo raciocínio, os resultados próximos à linha diagonal do gráfico (*Random Guess*) representam classificações completamente aleatórias.

A precisão (ACC) neste tipo de teste pode ser medido pela fórmula 3.2. TP representa os verdadeiros positivos enquanto que TN, o número de verdadeiros negativos. P e N são usados para representar o total de casos positivos e negativos, respectivamente.

$$ACC = \frac{TP + TN}{P + N} \tag{3.2}$$

Primeiramente, o analisador procura e marca todos e FE e LU. O primeiro teste trata da avaliação de cada um dos elementos do frame. É preciso saber se as unidades lexicais e os FEs foram corretamente identificados para depois avaliar se os frames como um todo foram corretamente classificados.

Para este teste foram analisadas 201 amostras. O resultado está representado na figura 3.18. A maioria dos elementos obteve bons resultados com a taxa de falsos positivos abaixo de 4% e verdadeiros positivos acima de 85%. A exceção ficou por conta dos elementos não nucleares *City* e *Date*. Mesmo assim a taxa de falsos positivos foi baixa (menor que 1%). A tabela 3.6 mostra a taxa de verdeiros positivos (TPR), falsos negativos (FPR) e a precisão (ACC).

O segundo teste analisa a classificação dos frames. Foram analisadas 546 amostras, onde cada sentença foi ranqueada pela rede Bayesiana com uma probabilidade de 0 a 100% de ser ou não um frame de oferta de trabalho. É importante lembrar que este

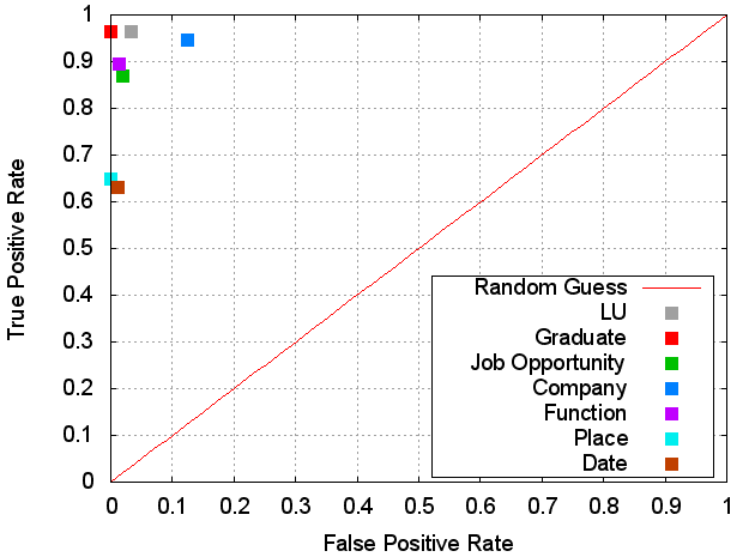

Figura 3.18: Gráfico ROC para FEs e LU.

Elemento	*TPR*	*FPR*	*ACC*
LU	0,964	0,033	0,965
Graduate	0,964	0,001	0,995
Job Opportunity	0,869	0,020	0,925
Company	0,946	0,125	0,940
Function	0,894	0,013	0,965
Place	0,649	0,001	0,866
Date	0,630	0,011	0,940

Tabela 3.6: TPR, FPR e ACC para FEs e LU.

conunto de amostras é disjunto do conjunto de sentenças que foi utilizado para realizar o treinamento da Rede bayesiana. Foram analisados vários limites para decidir à partir de qual valor uma classificação passaria a ser considerada como positiva. A figura 3.19 mostra o gráfico ROC considerando limites de 10%, 30%, 50%, 70% e 90%. Os limites marcados em 50% e 70% ficaram mais próximos do ponto $(0, 1)$ no gráfico. Dentre as 546 amostras, forma encontradas 64 evidências. A tabela 3.7 mostra a taxa de verdadeiros positivos (TPR), falsos negativos (FPR) e a precisão (ACC).

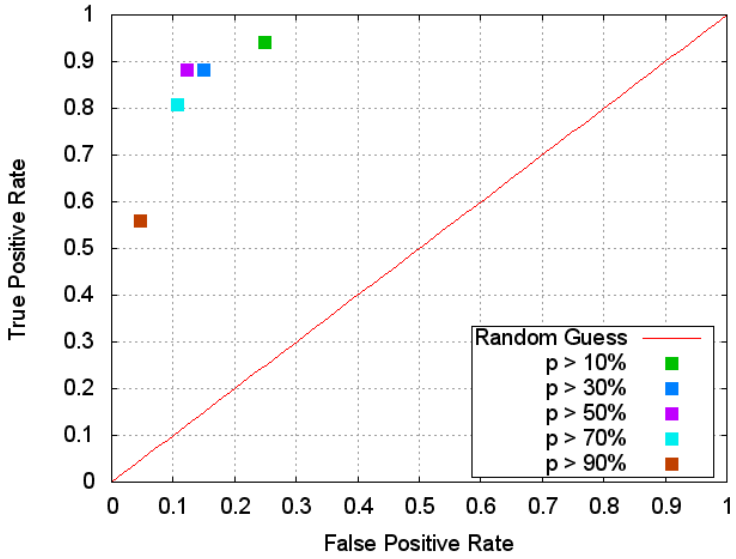

Figura 3.19: Gráfico ROC para classificação das sentenças.

Valor do teste	*TPR*	*FPR*	*ACC*
p > 10%	0,941	0,249	0,775
p > 30%	0,882	0,151	0,853
p > 50%	0,882	0,123	0,877
p > 70%	0,809	0,107	0,883
p > 90%	0,559	0,046	0,905

Tabela 3.7: TPR, FPR e ACC para classificação das sentenças.

3.3.11 Conclusões

Este trabalho propôs um modelo multiagente para fazer acompanhamento da evolução profissional de graduados de instituições de ensino superior. Esta é uma tarefa que envolve a busca de informações em diversas fontes e em diversos formatos, exigindo o uso de técnicas de processamento de linguagem natural e de sistemas distribuídos. Sendo assim, o uso de sistemas multiagentes dotados de técnicas de PLN surge como uma alternativa adequada. O sistema foi implementado segundo essa filosofia e usa diversas técnicas para extrair informações sobre os egressos. Alguns agentes são mais complexos que outros. No entanto, até mesmo os mais simples, como o agente que envia questionários, contribui com informações que são armazenadas no repositório de dados. Um dos agentes utiliza, além das técnicas típicas de processamento de linguagens

naturais, tais como um anotador sintático, ontologias e frames semânticos. Os Frames foram utilizados para caracterizar a cena que fornece a semântica de fundo para o enunciado. As ontologias foram usadas para definir a natureza dos conceitos associados aos itens lexicais. Os resultados se mostraram satisfatórios para uma primeira versão. A arquitetura do sistema permite que novos frames sejam utilizados para monitorar outras situações relacionadas com o acompanhamento de egressos.

Um dos problemas encontrados foi a necessidade de se manter uma base de nomes de locais e empresas na ontologia para realizar a identificação no corpo das sentenças. Uma alternativa seria a criação de um agente especializado que permanentemente busque essas informações na Internet. Esta tarefa seria uma sugestão para trabalhos futuros.

Também como sugestão de trabalho futuro, fica a oportunidade de se explorar mais as relações de inter Frames, já inseridas na ontologia. Através destas relações pode-se realizar buscas mais detalhadas com frames específicos, sendo que essa busca pode ser delegada para um outro agente. Por exemplo, pode-se usar um agente associado ao frame de localização para identificar cidades no texto. Este agente devolveria ao agente de oferta de trabalho as informações inerentes a localização, tirando essa a carga de trabalho do agente *contratante*.

É importante ressaltar que existe uma questão ética na coleta de informações sobre pessoas que tem sido objeto de acirrado debate. Em função disso, a efetiva implantação deste sistema deverá contar com o consentimento dos graduados e o sigilo das informações deve ser mantido.

Capítulo 4

Conclusões Gerais

Este trabalho propôs um modelo de sistema multiagentes para acompanhamento de egressos. A adoção de uma plataforma multiagente se revelou interessante uma vez que este é um problema complexo que necessita do uso de várias técnicas que podem ser empregadas de modo paralelo e distribuído para se alcançar os resultados. Além disso, o sistema multiagente que foi criado deixa portas abertas para novos incrementos no futuro, implementando, dessa forma, uma sistema escalável e adaptável.

Todo os objetivos desta pesquisa foram alcançados. Uma ontologia foi utilizada com frames semânticos para detectar evidências de relação empregatícia entre um egresso e alguma instituição. A descoberta de relações em documentos textuais é um tópico de pesquisa muito atual, como mostra uma recente publicação no Journal of the Brazilian Computer Society realizada por Abreu et al. [2013]. Para classificar de forma numérica as evidências encontradas, foi criada uma rede Bayesiana. Esta rede utiliza as informações fornecidas pela ontologia e o frame para encontrar a probabilidade de uma certa pessoa trabalhar ou ter trabalhado em uma dada empresa.

Foi realizado um estudo de caso com os ex-alunos de computação da Universidade Federal de Viçosa. Os dados foram extraídos de uma lista de discussão específica para esse grupo de egressos. Os testes realizados mostraram que é possível acompanhar a evolução profissional dos egressos utilizando múltiplos agentes, com soluções variadas da área de inteligência artificial e de processamento de linguagem natural. Particularmente

neste estudo: frames semânticos, ontologias e redes Bayesianas.

Como sugestão de trabalho futuro pode-se indicar um aperfeiçoamento na parte do sistema que tenta identificar referências às cidades. É também possível tirar proveitos de sites de busca para identificar elementos do frame, eliminando a necessidade de se incluir indivíduos em uma ontologia. Fica também como sugestão a ideia de se utilizar os diversos relacionamentos Frame-Frame para se tentar identificar padrões de forma mais granular e também como forma de se delegar o processamento para diversos agentes.

Referências Bibliográficas

Abreu, S., Bonamigo, T., and Vieira, R. (2013). A review on relation extraction with an eye on portuguese. *Journal of the Brazilian Computer Society*, 19(4):553–571.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4):50–58.

Artiles, J., Gonzalo, J., and Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of SemEval 2007 Workshop*, Prague, Czech Republic. Association of Computational Linguistics (ACL).

Austin, J. L. (1975). *How to do things with words*, volume 1955. Oxford university press.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. Montreal, Canada. COLING-ACL '98.

Balog, K. (2008). *People Search in the Enterprise*. Phd thesis, University of Amsterdam.

Balog, K., Azzopardi, L., and De Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM.

Balog, K., Azzopardi, L., and de Rijke, M. (2009). Resolving person names in web people

search. *in King I. and Ricardo Baeza-Yates R., (eds.) Weaving Services, Locations, and People on the WWW, Springer*, pages p. 301–323.

Balog, K. and de Rijke, M. (2008). Associating people and documents. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR 2008)*, pages p. 296–308.

Bateman, J. A., Magnini, B., and Fabris, G. (1995). The generalized upper model knowledge base: Organization and use. *Towards very large knowledge bases*, pages 60–72.

Ben-Gal, I. (2007). Bayesian networks. *Encyclopedia of statistics in quality and reliability*.

Bernon, C., Cossentino, M., and Pavón, J. (2005). Agent-oriented software engineering. *The Knowledge Engineering Review*, 20(02):99–116.

Bolzan, W. and Giraffa, L. M. M. (2002). Estudo comparativo sobre sistemas tutores inteligentes multiagentes web. Technical report, FACIN-PUCRS, Porto Alegre, Brazil. 54 p.

Bordini, R. H., Braubach, L., Dastani, M., Seghrouchni, A. E. F., Gomez-Sanz, J. J., Leite, J., O'Hare, G., Pokahr, A., and Ricci, A. (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica (03505596)*, 30(1).

Bordini, R. H., Dastani, M., and Winikoff, M. (2007). Current issues in multi-agent systems development. In *Engineering Societies in the Agents World VII*, pages 38–61. Springer.

Bordini, R. H., Vieira, R., and Moreira, A. F. (2001). Fundamentos de sistemas multiagentes. In *Jornada de Atualização em Informática (JAI' 01)*, Fortaleza, Brazil.

Carrasco, R. S., Oliveira, A. P., Lisboa, J., Moreira, A., and Arroyo, J. E. (2011). Linguistic structures to support an evidence tracking system for wildlife trafficking. CLEI 2011.

Chaib-draa, B. and Dignum, F. (2002). Trends in agent communication language. *Computational intelligence*, 18(2):89–101.

Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999). What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, 14(1):20–26.

Craswell, N. and de Vries, A. (2006). Overview of the trec-2005 enterprise track. *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*.

Elmacioglu, E., Tan, Y. F., Yan, S., Kan, M., and Lee, D. (2007). Psnus: Web people name disambiguation by simple clustering with rich features. Prague, Czech Republic. Association of Computational Linguistics (ACL).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):p. 861–874. ROC Analysis in Pattern Recognition.

Fillmore, C. J. (1977). Scenes-and-frames semantics. *Linguistic structures processing*, 59:55–88.

Fillmore, C. J. (2006). Frame semantics. *Cognitive linguistics: Basic readings*, pages 373–400.

Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pages 456–463. ACM.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:p. 131–163.

Genesereth, M. R., Fikes, R. E., et al. (1992). Knowledge interchange format-version 3.0: Reference manual.

Giunchiglia, F., Mylopoulos, J., and Perini, A. (2002). The tropos software development methodology: Processes, models and diagrams. Bologna, Italy. AAMAS Conference.

Guerra-Hernández, A., El Fallah-Seghrouchni, A., and Soldano, H. (2005). Learning in bdi multi-agent systems. In *Computational logic in multi-agent systems*, pages 218–233. Springer.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):p. 10–18.

Heckerman, D., Mamdani, A., and Wellman, M. P. (1995). Real-world applications of bayesian networks. *Communications of the ACM*, 38(3):24–26.

Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. Technical report, FURB, Blumenau, Brazil.

Jensen, F. V. (1996). *An introduction to Bayesian networks*, volume 210. UCL press London.

Khan, M. U. and Khan, S. A. (2009). Social networks identification and analysis using call detail records. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pages 192–196. ACM.

McCarthy, T. R. (2001). Don't fear carnivore: It won't devour individual privacy. *Mo. L. Rev.*, 66:827.

Meurs, M.-J., Lefevre, F., and Mori, R. D. (2009). Spoken language interpretation: On the use of dynamic bayesian networks for semantic composition. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages p. 4773–4776.

Minsky, M. (1974). A framework for representing knowledge.

Moreira, A. (2012). *An Ontology Grounded Framework for Frames Detection*. Doctor thesis, Federal University of Juiz de Fora, Brazil.

Moreira, A. and Salomão, M. M. M. (2013). Application of bayesian networks and ontological types to a lexeme to estimate its relevance to a semantic frame. *Veredas: Frame Semantics and Its Technological Applications*, 17(1):149–164.

Odell, J., Parunak, H. V. D., and Bauer, B. (2000). Extending uml for agents. *Ann Arbor*, 1001:48103.

Pearl, J. and Russel, S. (2000). Bayesian networks. *Handbook of Brain Threory and Neural Networks*, pages 157–160.

Petruck, M. R. (1996). Frame semantics. *Handbook of pragmatics*, pages 1–13.

Popescu, O. and Magnini, B. (2009). Alleviating the problem of wrong coreferences in web person search. In *The 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages p. 280–293, Mexico City, Mexico. CICLing 2009.

Reed, S. L., Lenat, D. B., et al. (2002). Mapping ontologies into cyc. In *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, pages 1–6.

Rodrigues, D. F., Oliveira, A. P., Filho, J. L., and Moreira, A. (2012). Semi-automatic follow-up of graduates. XXXI International Conference of the Chilean Computer Science Society (SCCC 2012).

Ruppenhofer, J., Petruck, M. E. M. R. L., Johnson, C. R., and Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice*.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. (1995). *Artificial intelligence: a modern approach*, volume 2. Prentice hall Englewood Cliffs.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Scheffczyk, J., Baker, C. F., and Narayanan, S. (2006a). Ontology-based reasoning about lexical resources. In Oltramari, A., editor, *ONTOLEX 2006*, pages p. 1–8, Genoa, LREC.

Scheffczyk, J., Pease, A., and Ellsworth, M. (2006b). Linking framenet to the suggested upper merged ontology. In Bennett, Brandon; Fellbaum, C., editor, *Formal Ontology in Information Systems (FOIS-2006)*, pages p. 289–300. IOS Press.

SemEval, editor (2007). *Proceedings of SemEval 2007 Workshop*, Prague, Czech Republic. Association of Computational Linguistics (ACL).

Tognini-Bonelli, E. (2001). *Corpus linguistics at work*, volume 6. John Benjamins Publishing.

Uschold, M. and Gruninger, M. (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11:p. 93–136.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons Inc.

Wooldridge, M., Jennings, N. R., et al. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152.

Zambonelli, F. and Omicini, A. (2004). Challenges and research directions in agent-oriented software engineering. *Autonomous Agents and Multi-Agent Systems*, 9(3):253–283.

Apêndice A

Armazenamento da Ontologia em Banco de Dados Relacional

Utilizamos uma Ontologia para formalizar todos os conceitos envolvidos no problema de acompanhamento de egressos. Com esta ferramenta, pudemos definir todos os termos e relações existentes. Além disso, foi possível aplicar lógica para fazer inferências sobre os indíviduos da ontologia e extrair ainda mais informações. Foi usada a ferramenta Protégé[1] para modelar toda a ontologia.

No entanto, foram encontrados alguns problemas para processar as informações. Notadamente, problemas de performance. O que acontece é que as ontologias são armazenas em arquivos de texto, no formato XML, e, quando o número de indivíduos começa a ficar grande, o uso das ferramentas de ontologias convencionas torna-se inviável. Somente os indivíduos que representam os municípios do Brasil são cerca de 5570. Um arquivo de texto não possui mecanismos para lidar com um armazenamento deste porte, diferente do que acontece com sistemas gerenciadores de bancos de dados.

Visto que seria inviável trabalhar com uma ontologia com tantos indíviduos, optamos por uma solução alternativa. A ideia é manter todas as classes e relações na ontologia e armazenar os indivíduos em um banco de dados relacional. Os indivíduos seriam carregados para dentro da ontologias à medida que forem necessários. Desta forma,

[1] http://protege.stanford.edu/

podemos contornar o problema de performance sem perder os recursos de lógica e inferência.

O modelo relacional está representado na Figura A.1. Foi utilizado o Diagrama de Classe UML e o esteriótipo *«Table»* foi usado para representar as tabelas. As relações representam chaves estrangeiras, onde a classe que possui o valor * referencia um indivíduo da classe da outra extremindade, marcada com *1*.

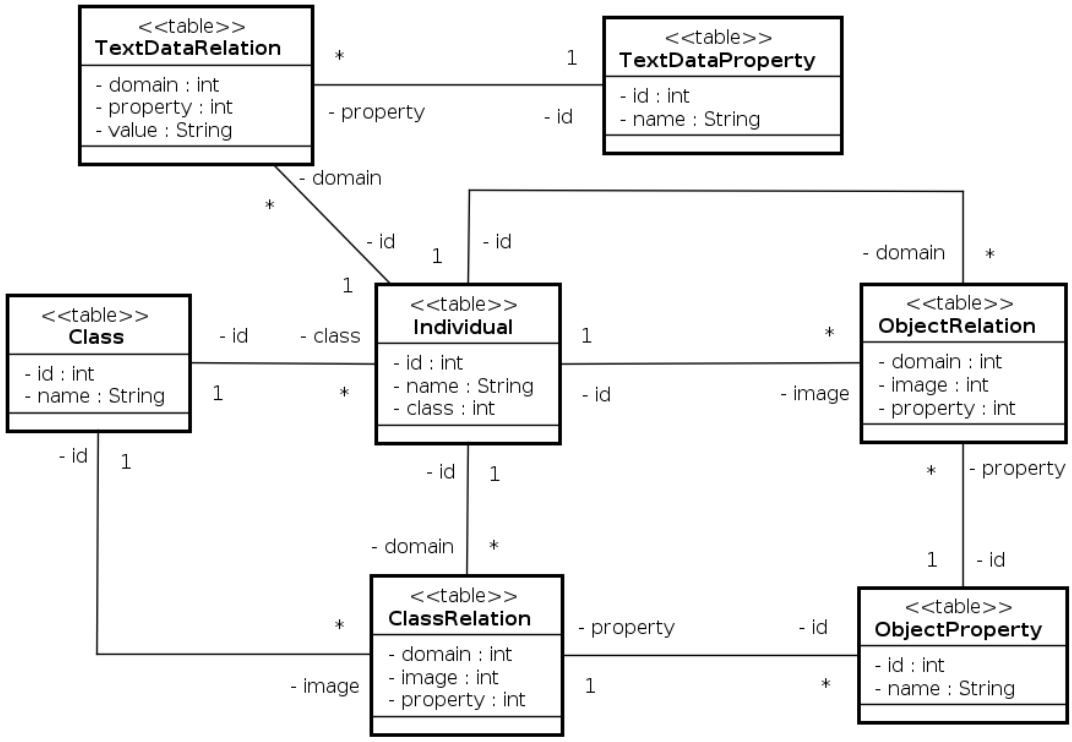

Figura A.1: Modelo da Ontologia em Banco Relacional.

A tabela *Class* armazena as classes da ontologia. Mas apenas as classes cujo indivíduos estão armazenados no banco. As tabela *Individual*, faz referência à tabela *Class*, representado à classe a qual pertence.

As relações da ontologias foram mapeadas para a tabela *ObjectProeprty*. A tabela *ClassRelation* representa relações entre uma classe e um indivíduo. De forma parecida, a tabela *ObjectRelation* representa relações entre dois indivíduos.

As properiedades são representadas pela tabela *TextDataProperty*. Foi usada apenas propriedades do tipo texto, mas pode-se extender para os demais tipos (inteiro, decimal,

caractere, etc). As instanciações de propriedades dos indivíduos estão armazenadas na tabela *TextDataRelation*.

À medida que os indivíduos são requisitados na ontologia, eles são carregados do banco de dados. Em consequência, todas as relações e propriedades deste indíviduo também são carregados, assim como os outros indíviduos, que estão na outra ponta da relação.