

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**Desenvolvimento de um assistente pessoal
de viagem sobre um modelo FrameNet dos
domínios de Turismo e Esportes**

Arthur Lorenzi Almeida

JUIZ DE FORA
JUNHO, 2017

Desenvolvimento de um assistente pessoal de viagem sobre um modelo FrameNet dos domínios de Turismo e Esportes

ARTHUR LORENZI ALMEIDA

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Sistemas de Informação

Orientador: Wagner Arbex

JUIZ DE FORA
JUNHO, 2017

DESENVOLVIMENTO DE UM ASSISTENTE PESSOAL DE
VIAGEM SOBRE UM MODELO FRAME.NET DOS DOMÍNIOS DE
TURISMO E ESPORTES

Arthur Lorenzi Almeida

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Wagner Arbex
Doutor em Engenharia de Sistemas e Computação

Fernanda Cláudia Alves Campos
Doutora em Engenharia de Sistemas e Computação

Victor Ströele de Andrade Menezes
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA
21 DE JUNHO, 2017

Aos meus amigos e irmãos.

Aos pais, pelo apoio e sustento.

Resumo

Dando continuidade às pesquisas recentes do laboratório FrameNet Brasil, o presente trabalho apresenta o m.knob (*Multilingual Knowledge Base*), uma aplicação móvel que disponibiliza, além de um tradutor e uma dicionária, um assistente pessoal de viagem capaz de recomendar atrações turísticas processando as entradas do usuário em linguagem natural. Esse sistema de recomendação é construído sobre uma base de conhecimento gerada a partir de uma FrameNet, para os domínios do Turismo e do Esporte, e de informações obtidas através do processamento de um *corpus* de comentários de plataformas online. O procedimento usado para a busca de atrações na base utiliza técnicas de *spreading activation*, onde a ativação é iniciada pelos elementos da sentença do usuário. Para a construção do *chatbot* que interage com o sistema de recomendação, foi desenvolvida uma arquitetura que permite a definição de fluxos de conversa extensíveis e que interage com outros serviços de processamento de linguagem natural disponíveis na Internet.

Palavras-chave: chatbot, framenet, recomendação.

Abstract

Following recent researches at FrameNet Brazil's laboratory, the present work presents m.knob (Multilingual Knowledge Base), a mobile application that provides, in addition to a translator and a dictiopedia, a personal travel assistant who can recommend tourist attractions based on user's input in natural language. This recommendation system is built over a knowledge base created from a Tourism and Sport domain FrameNet and from information obtained through the processing of a corpus of comments from online platforms. The procedure used to search for attractions in the database uses spreading activation techniques, initiating activation from the elements of the user's sentence. To develop the chatbot that interacts with the recommendation system, an architecture that allows the definition of extensible chat flows was developed. This architecture also provides ways for the system to interact with other natural language processing services available on the Internet.

Keywords: chatbot, framenet, recommendation.

Agradecimentos

Aos pesquisadores e bolsistas (de graduação, mestrado e doutorado) da FrameNet Brasil, que construíram diversas partes deste trabalho. Sem suas contribuições este não se realizaria.

Ao Wagner, pela orientação, amizade e todos os conhecimentos transmitidos ao longo da iniciação científica.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos.

Conteúdo

Lista de Figuras	6
Lista de Tabelas	7
Lista de Abreviações	8
1 Introdução	9
1.1 Apresentação do tema	9
1.2 Justificativa	10
1.3 Objetivos	11
1.4 Organização	12
2 Fundamentação teórica	13
2.1 Chatbots	13
2.2 Sistemas de recomendação	14
2.3 Semântica de frames e FrameNets	17
2.4 Estruturas qualia	19
3 Revisão da literatura	21
3.1 Uso de chatbots para a recomendação	21
3.2 Trabalhos anteriores da FN-Br	26
4 Arquitetura do m.knob	32
4.1 Interface de usuário	32
4.2 Interações com serviços externos	34
4.3 Controlador de diálogo	35
5 Sistema de recomendação	39
5.1 A rede do m.knob	40
5.1.1 Categorização de atrações	40
5.1.2 Incorporação de Frames aos papéis Qualia	41
5.1.3 Pesos das relações	42
5.2 Ranqueamento de atrações	43
5.3 Adaptação às críticas do usuário	45
6 Considerações finais	48
Bibliografia	49

Lista de Figuras

2.1	O frame <code>Obter_documento</code> na FN-Br.	18
2.2	Papéis qualia para <i>livro.n.</i>	19
3.1	Arquitetura multi estratégia do R-cube. Extraído de (KIM; BANCHS, 2014).	24
4.1	Componentes da arquitetura do m.knob.	32
4.2	Telas do aplicativo em português.	33
4.3	Exemplo de definição das falas do <i>chatbot</i>	36
4.4	Exemplo de configuração de roteiro do <i>chatbot</i>	37
5.1	Ciclo de <i>feedback</i> do sistema de recomendação do m.knob.	39
5.2	Relações entre nós do grafo do m.knob. Arcos em vermelho possuem peso 0,5.	42
5.3	Curva da Equação 5.2.	44
5.4	Interação bem sucedida com o sistema de recomendação através do <i>chatbot</i>	45
5.5	Exemplo de ciclo de <i>feedback</i> do sistema de recomendação.	47

Lista de Tabelas

3.1	Exemplo de vetor de característica representando quatro características: Compras, <i>Fast-food</i> , Café e <i>Rugby</i> (SILVA et al., 2017).	29
5.1	Relações qualia ternárias na base do m.knob.	41

Lista de Abreviações

API	<i>Application Programming Interface</i>
DCC	Departamento de Ciência da Computação
ECA	<i>Embodied conversational agents</i>
EF	Elementos de frame
FN	FrameNet
PLN	Processamento de linguagem natural
RI	Recuperação de Informação
UL	Unidade lexical
TLG	Teoria do Léxico Gerativo
UFJF	Universidade Federal de Juiz de Fora

1 Introdução

1.1 Apresentação do tema

A popularização da Web a partir da década de 90, aliada ao crescimento da Internet, estimulou o desenvolvimento de muitas novas tecnologias, dentre essas tecnologias, destacam-se os sistemas de recomendação. Os primeiros sistemas de recomendação foram criados para realizar a filtragem da informação com o auxílio de dados informados por seres humanos. Essa classe de sistemas porém, não se restringe exclusivamente a tal técnica. Atualmente, a recomendação pode auxiliar um usuário a encontrar de maneira mais fácil o que deseja ou oferecer ativamente um item que possa lhe interessar. Em ambas as formas, os sistemas de recomendação estão presentes nos tipos de serviços mais populares da Web: plataformas de *e-commerce*, provedores de conteúdo, mídias sociais e redes sociais. A valorização desses sistemas por parte das empresas líderes de mercado em suas respectivas áreas de atuação evidencia sua importância. A Netflix chegou a promover uma competição pelo melhor algoritmo para a predição de avaliações de filmes ¹. No Website da Amazon, a seção “Clientes que compraram este item também compraram” é bem conhecida. Tanto a Netflix quanto o Youtube apresentam recomendações para o usuário em sua página inicial, enquanto o Facebook frequentemente sugere possíveis amizades.

As implementações mais conhecidas de sistemas de recomendação geralmente oferecem ao usuário alguns itens baseando-se em dados históricos de maneira ativa, porém, esses sistemas também são muito úteis em casos nos quais a recomendação atende uma requisição explícita do usuário. Nesses casos, além de ser capaz de recomendar, o sistema deve ser capaz de interpretar a requisição do usuário. Para isso pode utilizar formulários, uma sintaxe específica ou a linguagem natural. Essas formas de interação entre um ser humano e um computador são bastante diferentes quando consideradas, por exemplo, a complexidade de desenvolvimento, as limitações à expressividade do usuário e a curva de aprendizagem da ferramenta. Sistemas que permitem o uso de linguagem natural não

¹<https://netflixprize.com/>

limitam a entrada do usuário nem exigem esforço do usuário para aprender, porém, seu desenvolvimento é extremamente desafiador e por isso, tópico constante de pesquisa. A interação humano-computador, através da linguagem natural, normalmente é feita através de interfaces conversacionais. Essas interfaces dependem da existência de um agente que deve interpretar as entradas do usuário e realizar as tarefas exigidas no sistema. Esses agentes interativos são conhecidos como *chatbots*, *chatterbots*, *talkbots* ou apenas *bots* e recentemente têm ganhado maior interesse por parte da comunidade científica e da indústria devido aos avanços e popularização da Inteligência Artificial. Além disso, *chatbots* podem ser facilmente integrados a aplicações existentes através de APIs para a realização de tarefas variadas, incluindo as de sistemas de recomendação.

As técnicas utilizadas na construção de *chatbots* vão de simples casamentos de padrão ao aprendizado de máquina (KIM; BANCHS, 2014; NICULESCU et al., 2014; D’HARO et al., 2015), e podem envolver também o uso de ontologias (GARRIDO et al., 2016). Alternativamente, o processamento de linguagem natural pode ser realizado sobre uma base de conhecimento. O m.knob, *Multilingual Knowledge Base* (Base de Conhecimentos Multilíngue), é um projeto iniciado pelo laboratório de Linguística Computacional FrameNet Brasil (FN-Br) com o objetivo de utilizar o processamento semântico de línguas naturais para oferecer uma experiência turística inovadora. A base de conhecimento do m.knob é construída sobre uma FrameNet (um banco de dados léxico que armazena palavras e suas relações com *frames* semânticos) desenvolvida pela FN-Br. Uma das funcionalidades oferecidas pelo m.knob é a recomendação de atrações turísticas a partir dos pedidos do usuário através de linguagem natural (COSTA et al., 2018). Essa funcionalidade pode ser aperfeiçoada com a utilização de um agente interativo que garante uma interface consistente, melhorando a experiência do usuário.

1.2 Justificativa

Ao planejar uma viagem, turistas precisam de informações específicas sobre as atrações que planejam visitar. As informações contidas em guias turísticos, apesar de úteis, não abrangem todos os aspectos importantes para a experiência turística ideal de um indivíduo ou grupo. Para alguns turistas, locais românticos serão mais valorizados, enquanto para

outros a capacidade de levar seu animal de estimação ou ter contato com a natureza pode ser mais importante. Guias escritos geralmente limitam a descrição a características mais gerais de um grupo seletivo de atrações. Esses problemas não são frequentes porém em plataformas online que possuem informações turísticas. Essas aplicações geralmente permitem que os próprios usuários escrevam avaliações sobre os mais variados locais de interesse, mas introduzem outro problema: sobrecarga de informação. Navegar por milhares de comentários de diferentes atrações é demorado e prejudica a experiência do usuário final.

Para endereçar o problema da sobrecarga de informação, este trabalho apresenta um sistema de recomendação para o m.knob através do uso da base FrameNet e um *chatbot*. O sistema facilita o processo de filtragem de atrações turísticas baseando-se em informações gerais e específicas. O uso de uma interface conversacional está alinhado com o conhecimento de características específicas dos locais, isto é, as características relevantes descritas pelos usuários não são conhecidas *a priori* e por isso a interface não deve restringir as entradas do usuário.

1.3 Objetivos

Este trabalho tem como objetivo principal o desenvolvimento do m.knob app: uma aplicação móvel que entregue uma experiência turística diferenciada. Para isso, os seguintes objetivos específicos foram enumerados:

1. Rever trabalhos anteriores da FN-Br sobre o domínio do turismo;
2. Projetar uma arquitetura para a integração das diferentes funcionalidades ou serviços do m.knob;
3. Propor um sistema de recomendação de atrações turísticas sobre uma base de conhecimento FrameNet;
4. Desenvolver uma aplicação móvel com uma interface conversacional para a recomendação;

1.4 Organização

O presente trabalho está dividido em 6 Capítulos. O primeiro Capítulo contextualiza o trabalho introduzindo o tema tratado e apresentando suas justificativas e objetivos. O Capítulo 2 trata da fundamentação teórica deste trabalho, parte importante em um trabalho multidisciplinar. Os conceitos de *chatbot*, Sistemas de Recomendação são apresentados juntamente com a Semântica de Frames (teoria no qual se baseiam as FrameNets) e as estruturas Qualia. A revisão da literatura é apresentada no Capítulo 3. O Capítulo 5, descreve o sistema de recomendação do m.knob, o banco de dados utilizado, o algoritmo utilizado para a identificação dos locais a serem recomendados e o endereçamento das críticas do usuário. A arquitetura geral do m.knob app é apresentada no Capítulo 4, onde são descritas as interações entre os serviços da FN-Br, serviços de PLN como Wit.ai e a interface conversacional da aplicação. O Capítulo 6 por fim, discute possíveis melhorias para o projeto a serem estudadas em futuros trabalhos.

2 Fundamentação teórica

2.1 Chatbots

Chatbots são programas que interagem com clientes humanos através da linguagem natural, da maneira que um humano faria. Essa interação geralmente tem um propósito específico relacionado a tarefa para a qual o robô foi projetado para realizar. Os principais desafios no desenvolvimento de um *chatbot* estão ligados à compreensão da entrada do usuário e do contexto da conversa que está acontecendo (BRADEŠKO; MLADENIĆ, 2012).

Recentemente, o interesse da comunidade científica e de empresas por *chatbots* tem aumentado, principalmente devido aos avanços em *hardware* e nas áreas de Inteligência Artificial e Processamento de Linguagem Natural. O tópico, porém, não é recente. Desenvolvido em 1966, ELIZA foi o primeiro *chatbot* conhecido e se comportava como um psicoterapeuta. As respostas eram construídas através da identificação de palavras-chave e de um contexto mínimo, que era utilizado para determinar qual regra de resposta seria utilizada. Essas respostas eram formadas pelas regras e uma transformação da entrada do usuário. Para a entrada “Eu fiz um longo passeio de barco” seria dada a resposta “Fale mais sobre barcos”, que não soa natural na maioria dos contextos, porém, como psicoterapeuta, ELIZA poderia assumir uma posição de não conhecimento do mundo real sem prejudicar a interação com um ser humano (WEIZENBAUM, 1966).

Desde o trabalho de Weizenbaum, muitos outros *chatbots* foram criados para passar no Teste de Turing. O Loebner Prize é a competição mais famosa relacionada a *chatbots*. Nela juízes desenvolvem conversas com *chatbots* e premiam aquele no qual a conversa mais se aproxima a uma com um humano ². Porém, as técnicas utilizadas pouco contribuem para os campos da Compreensão da Linguagem Natural e Inteligência Artificial.

Os trabalhos mais recentes relacionados a *chatbots* têm dado importância a inter-

²<https://loebner.exeter.ac.uk/>

pretação das entradas do usuário com técnicas mais precisas para a construção de agentes melhores. Neste trabalho, por exemplo, com o uso de uma FrameNet espera-se que o *bot* possa interpretar entradas muito mais complexas devido à presença de informação semântica.

2.2 Sistemas de recomendação

Sistemas de recomendação são ferramentas que recomendam a um usuário itens que são de seu interesse (RICCI; ROKACH; SHAPIRA, 2011). O termo “item” é usado na literatura para denotar o que o sistema recomenda e pode se referir a produtos, mídias ou conexões (redes sociais), por exemplo. Uma das formas pelas quais um sistema define quais itens recomendar consiste em inferir como o usuário avaliaria os itens conhecidos, baseando-se na correlação entre as atividades relacionadas aos itens e aos usuários. O desafio de prever como um usuário avaliaria um determinado item é usado por (AGGARWAL, 2016) para definir o problema da recomendação: assume-se a existência de um conjunto de treinamento que indica a preferência de alguns usuários por determinados itens. Para m usuários e n itens, isso corresponde a uma matriz $m \times n$ onde os valores observados são utilizados para o treinamento. Os valores faltantes devem ser preditos por um modelo. Uma outra formulação do problema da recomendação (sendo essa mais específica) consiste em determinar os k -melhores itens a serem recomendados para um usuário através de um ranqueamento (AGGARWAL, 2016).

Além de recomendar itens que são considerados relevantes para o usuário, sistemas de recomendação possuem outros objetivos operacionais que devem ser alcançados:

1. *Novidade (novelty)*: O sistema se torna mais útil quando recomenda itens que o usuário ainda não conhece.
2. *Serendipidade (Serendipity)*: As recomendações devem ser, até certo ponto, inesperadas. Para que um item seja uma novidade, basta que o usuário não o conheça, porém para ser um “acaso” é necessário que ele seja diferente do que o usuário espera das recomendações. Por exemplo, considerando um sistema de recomendação de músicas e um usuário que gosta de *jazz*, recomendar um álbum de *jazz* recém

lançado é novidade, mas um álbum de *reggae* é acaso. Ocasionalmente, os itens nesse tipo de recomendação são irrelevantes, mas a longo prazo essas recomendações são benéficas porque dão ao usuário a oportunidade de descobrir novas categorias de itens que lhe interessam.

3. Diversidade (*diversity*): Recomendar itens muito similares, mesmo se tratando dos k -melhores, aumenta o risco do usuário achar todos os itens irrelevantes. A diversidade garante esse benefício a curto e a longo prazo (no caso em que o mesmo tipo de item é recomendado várias vezes).

Os modelos de sistema de recomendação podem ser classificados em três tipos gerais: colaborativos, baseados em conteúdo ou baseados em conhecimento (e híbridos). Os dois primeiros tipos são mais comuns e compartilham mais características entre si que o terceiro (AGGARWAL, 2016).

Sistemas de recomendação colaborativos trabalham sobre a ideia na qual geralmente as avaliações observadas são altamente relacionadas entre vários usuários e itens. Através de uma métrica de similaridade, o sistema pode utilizar usuários similares (vizinhança) para inferir as avaliações faltantes na matriz esparsa de avaliações de usuários sobre itens. O uso de métodos baseados em vizinhanças não é condição necessária para que um sistema de recomendação seja colaborativo. Métodos de aprendizado de máquina podem ser utilizados para a inferência e se apresentam como alternativas mais eficazes, apesar de serem mais complexos, menos eficientes e possivelmente apresentarem problemas de justificabilidade dos resultados (DESROSIERS; KARYPIS, 2011; KOREN; BELL, 2015).

Uma maneira diferente de se recomendar itens consiste em identificar as preferências de um usuário para determinadas características de itens. Sistemas com essas características são chamados de sistemas de recomendação baseados em conteúdo (*content-based*). Esse tipo de sistema geralmente é composto de três componentes principais: um analisador de conteúdo, que constrói os vetores de atributos dos itens a serem recomendados; o “aprendiz de perfil”, que generaliza os dados que representam preferências do usuário e componente de filtragem que casa o perfil do usuário com itens, utilizando os atributos conhecidos. As principais vantagens de sistemas de recomendação baseados em

conteúdo são a independência entre usuários e a transparência do conhecimento, já que estrutura-se os atributos dos itens e dos perfis de usuário. Além do mais, são capazes de recomendar itens que nenhum usuário avaliou porque conhecem seus atributos, diferentemente de sistemas colaborativos que dependem de avaliações. Apesar disso, o problema de partida fria ainda está presente nos sistemas de recomendação baseados em conteúdo porque ainda há a necessidade de se aprender o perfil de um novo usuário. Outro problema desse tipo de sistema é a falta de serendipidade e possivelmente de novidade. Essa superespecialização ocorre porque apenas o perfil do próprio usuário é considerado, efetivamente limitando-o as suas próprias preferências. Por fim, a análise dos itens pode não ser capaz de capturar todos os aspectos que podem vir a ser de interesse do usuário, principalmente em domínios considerados complexos (LOPS; GEMMIS; SEMERARO, 2011).

O terceiro grupo de sistemas de recomendação engloba aqueles baseados em conhecimento. Esses são especialmente úteis em contextos onde o usuário deseja especificar restrições à recomendação ou existe uma escassez de dados históricos de avaliações. Essas características são facilmente encontradas em sistemas que apoiam atividades realizadas com pouca frequência e/ou atividades em que a escolha do melhor item é muito importante para o usuário, como por exemplo, a aquisição de um imóvel. Por não dependerem de avaliações anteriores para realizar recomendações, esse tipo de sistema não apresenta o problema da partida fria, tornando-o conveniente para os casos em que o usuário deve receber boas recomendações a partir do momento em que começa a interagir com a aplicação. Essa capacidade não deriva apenas da base de conhecimento, mas também das críticas que o usuário pode fazer sobre as recomendações durante uma sessão de interação com o sistema. Essas críticas podem ser realizadas de várias maneiras de acordo com a interface da aplicação e adaptam os resultados retornados pelo sistema até que eles sejam satisfatórios. Arquiteturalmente, isso implica na presença de um *loop* de *feedback* em que o usuário gradualmente ajusta suas restrições (AGGARWAL, 2016; FELFERNIG et al., 2011).

A aplicação desenvolvida neste trabalho usa uma base de conhecimento (de locais, palavras e *frames*) para recomendar pontos de interesse. A construção da base é descrita

na Seção 5.1, a forma em que as críticas do usuário são endereçadas é discutida na Seção 5.3.

2.3 Semântica de frames e FrameNets

A Semântica de Frames é uma teoria que se baseia na hipótese na qual significado é construído considerando experiências culturais e conhecimento do mundo (FILLMORE et al., 1982). *Frames* são as estruturas que modelam esse conhecimento através de referências entre si, isto é, a compreensão do significado de um *frame* depende da compreensão de toda a superestrutura em que ele está inserido. Essas superestruturas são chamadas de FrameNets, inicialmente foram desenvolvidas pela Berkeley FrameNet (BFN) para o inglês, mas atualmente são desenvolvidas em vários outros idiomas. A FN-Br expande essa base de dados para o português.

Os *frames* de uma FrameNet são usados para descrever eventos, situações ou objetos e são constituídos de relações com outros *frames*, elementos de *frame* (EFs) e as unidades lexicais que evocam esse *frame*. As relações entre *frames* podem ser de oito diferentes tipos: herança, uso, perspectiva sobre, subframe, precede, inquisitivo de, causativo de, metáfora e uma relação “ver também”, utilizada apenas por humanos (RUPPENHOFER et al., 2016).

Os elementos de *frame* representam os participantes ou propriedades do *frame* em questão. Tomando como exemplo um *frame* para representar o ato de cozinhar, podemos considerar como EFs a pessoa que prepara o alimento, o alimento que é preparado, o recipiente onde o alimento é cozido e o instrumento utilizado que gera o calor necessário. Os elementos de *frame* usados como exemplo são chamados de nucleares, porque são essenciais para a composição do *frame*. Elementos não-nucleares são aqueles que acrescentam informações circunstanciais aos *frames* como propósito, maneira, momento, local, etc.

Unidades lexicais (UL) são os elementos básicos de uma língua e transmitem um significado único. Uma UL pode consistir de uma ou mais palavras. São exemplos de ULs *café.n*, *parque.n*, *jogar.v* e *beber.v* na notação comumente utilizada onde o lema é seguido de um ponto e uma etiqueta de parte do discurso (*Part-of-speech (POS) tagging*)

Obter_documento

Definição

O **Turista** busca obter os **Documentos** necessários para a realização de sua viagem.

Exemplo(s)

Elementos de Frame Nucleares

FE Core:

Documento [Document] Documento que o **Turista** deseja obter.

Turista [Tourist] Indivíduo ou grupo que deseja obter um dado **Documento**.

Elementos de Frame Não-Nucleares

Relações

É subframe de [Cenário_do_turismo_preparação](#)

É usado por [Documentos](#)

Unidades Lexicais

[obter.v](#) [renovar.v](#) [tirar.v](#)

Figura 2.1: O frame `Obter_documento` na FN-Br.

³. Nas FrameNets, as ULs relacionadas a um *frame* são armazenadas para que durante a análise de sentenças anotadas seja possível identificar os *frames* relacionados àquela sentença. Uma UL pode estar relacionada a vários *frames* diferentes e vice-versa. Isso significa que um *frame* não representa o significado de uma UL, mas o conhecimento no qual o significado da UL deve ser relativizado.

Além dos *frames*, EFs e ULs, uma FrameNet também é composta de um conjunto de anotações formado por sentenças e seus respectivos EFs, funções gramaticais e tipos de frases.

A Figura 2.1 apresenta o frame `Obter_documento` como exemplo.

³Em linguística, lema é a forma canônica de uma palavra que possui várias formas. O lema de "correr", "correndo" e "correu" é *correr.v*. Esse conjunto de variações que representa uma única unidade de significado é chamado de lexema.

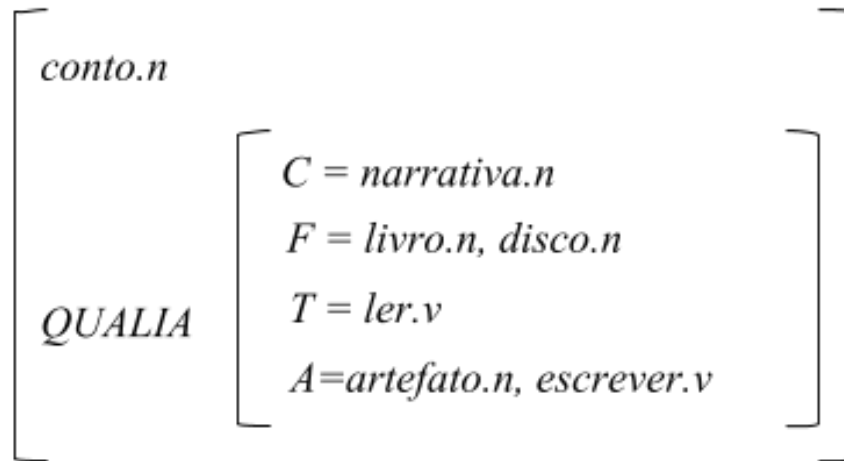


Figura 2.2: Papéis qualia para *livro.n*.

2.4 Estruturas qualia

As estruturas qualia constituem parte essencial da Teoria do Léxico Gerativo (TLG). Proposta em (PUSTEJOVSKY, 1991), a teoria contrapôs abordagens que consideravam que o significado das palavras poderia ser decomposto em um conjunto de traços sintáticos, semânticos e morfológicos, sugerindo o uso de um modelo estruturado onde a configuração semântica mínima de uma UL pudesse ser especificada para que o significado fosse gerado. Essa geração se daria através do uso de predicados lógicos de maneira similar a uma gramática onde a decomposição das palavras não seria utilizada para a identificação de traços, mas de dispositivos gerativos. Pustejovsky defendeu que o léxico gerativo, ao abandonar as características estáticas das abordagens anteriores, seria capaz de melhor representar fatores contextuais, facilitando a definição do significado sob domínios específicos.

A TLG assume o uso de quatro níveis de representação semântica para a obtenção do significado. Neste trabalho, porém, apenas as estruturas qualia serão consideradas porque são elas que definem as propriedades essenciais de uma UL e contribuem de maneira significativa para o adensamento de uma base FrameNet, essencialmente expandindo o número de relações entre ULs e o conhecimento representado pelos *frames*.

Para cada UL a ser especificada, a TLG define quatro papéis ou relações qualia. O primeiro quale é o constitutivo que relaciona um objeto com suas partes ou constituintes e é representado pela letra C. O quale formal, denotado F, estabelece o que distingue

o objeto em um contexto maior como por exemplo, sua orientação, magnitude, forma, dimensionalidade, cor e posição. O télico T, é a função ou propósito do objeto e pode definir também suas atividades. O último quale é o agentivo que representa os fatores envolvidos com a origem do objeto, como o criador, artefato, tipo natural ou sequência causal. O agentivo é identificado pela letra A. A Figura 2.2 apresenta, como exemplo, as relações qualia de *conto.n.*

A inclusão das relações qualia na base do m.knob é discutida nas seções 3.2 e 5.1.

3 Revisão da literatura

Neste Capítulo são apresentados trabalhos relacionados a este, dividindo-os em duas seções. A primeira Seção apresenta trabalhos nos quais *chatbots* foram usados para a recomendação de atrações turística e um onde restaurantes eram recomendados. A segunda Seção discorre sobre alguns trabalhos anteriores da FrameNet Brasil sobre o domínio do turismo, contextualizando esse trabalho e apresentando pesquisas que o tornaram possível.

3.1 Uso de chatbots para a recomendação

As aplicações desenvolvidas em trabalhos recentes com o objetivo de sugerir pontos de interesse geralmente possuem a característica de serem sensíveis ao contexto e algumas apresentam interfaces conversacionais (NICULESCU et al., 2014; D’HARO et al., 2015; KIM; BANCHS, 2014). Nesta Seção serão apresentados trabalhos da literatura que propõem sistemas para recomendação de atrações turísticas através de interfaces conversacionais.

O sistema de diálogo multimodal SARA, acrônimo para *Singapore’s Automated Responsive Assistant*, foi desenvolvido como uma aplicação móvel Android para oferecer assistência turística para visitantes de Singapura (NICULESCU et al., 2014). SARA é capaz de sugerir atrações locais, restaurantes, pontos turísticos, direções e serviços de transporte através de entradas por voz ou texto em inglês ou pela leitura de códigos QR.

O sistema é implementado sobre uma arquitetura cliente-servidor e os componentes do lado servidor são conectados utilizando o *framework* proposto por Jiang et al. (2014). O processamento de linguagem natural é feito em uma abordagem híbrida que considera regras e modelos estatísticos que foram treinados com 40 horas de diálogos envolvendo perguntas de turistas e respostas de guias.

O componente de controle de diálogos (*Dialog Manager*) foi desenvolvido utilizando abordagens baseadas em regras e exemplos onde as respostas são geradas baseando-se em *templates*. Um dos componentes da arquitetura é uma base de dados que armazena todos os dados que os outros componentes necessitam.

Nessa arquitetura, as principais funções do cliente Android são a leitura de códigos QR, a obtenção de informações de geolocalização e a conversão de fala em texto e vice-versa. Esses recursos são oferecidos nativamente pela plataforma Android.

Como parte final do trabalho, o sistema foi avaliado por 10 usuários em 3 cenários diferentes: o primeiro envolveu os aspectos de recomendação e informação do SARA em uma região específica da cidade, o segundo estava relacionado a obtenção de informações sobre locais em toda Singapura e direções para esses locais. O terceiro cenário tinha como objetivo avaliar a capacidade do sistema em responder informações gerais, como o clima e a moeda em Singapura. O sistema se comportou melhor no terceiro cenário (60% dos objetivos foram completados). O pior desempenho foi encontrado no segundo cenário, onde apenas um terço das tarefas foi completada com sucesso. Na avaliação dos usuários, a usabilidade e confiabilidade do sistema foram os pontos mais criticados. Segundo os autores, essas avaliações e problemas no segundo cenário decorrem do fato do sistema não ter identificado corretamente locais e direções.

O sistema CLARA, descrito pelos mesmos autores de SARA apresenta capacidades de busca com linguagem natural para dois domínios de aplicação diferentes: um sistema de informação para conferências e um guia turístico local (D'HARO et al., 2015).

No CLARA as respostas são geradas através de um processo que primeiramente verifica um índice de perguntas e respostas. Se a pergunta está no índice, o sistema usa a resposta já conhecida. Quando a pergunta não pode ser encontrada, o CLARA determina se usará o motor de busca de artigos ou de informações turísticas, a forma em que essa decisão é tomada não é descrita de maneira detalhada pelos autores. Para identificar as entidades no domínio de conferências um algoritmo de busca *fuzzy* é utilizado e, segundo os autores, esse algoritmo possui certa robustez a erros ortográficos. No motor de informações turísticas, um classificador de texto é usado para identificar o tópico de busca, depois o mesmo algoritmo *fuzzy* identifica as entidades no contexto de turismo.

Além das diferenças na forma de compreensão das entradas, a abordagem de D'Haro et al. (2015) se diferencia de Niculescu et al. (2014) na obtenção de informações. O CLARA usa a *web* para obter informações. Os autores afirmam que um *site* de restaurantes de Singapura é usado para a busca das informações após o sistema determinar as

recomendações baseado na localização do usuário.

O CLARA foi avaliado durante a *15th Conference of the International Speech Communication Association*. Para isso, as informações da conferência tiveram que ser obtidas previamente. Os resultados da avaliação demonstraram que a maior parte das *queries* foram de interações com o *bot*. Os autores afirmam que o sistema foi mais usado para interações com o CLARA e informações turísticas. Além disso, eles explicam alguns problemas que podem ter causado a baixa utilização para obtenção de informações da conferência: o aplicativo oferecia capacidade de busca incorporada, o mecanismo de recomendação que notificava o usuário e alguns jogos relacionados às funcionalidades de *chat* patrocinados pela organização do evento. Além desses resultados, os autores revelam que cerca de 50% das *queries* de usuários não foram respondidas pelo sistema, porém 75% dessas eram de interações fora do domínio da aplicação em que o sistema não conseguiu encontrar respostas no índice interno nem na *web*.

Agentes conversacionais para a recomendação também são encontrados na literatura para domínios relacionados. É o exemplo de Kim e Banchs (2014) que apresenta o R-cube: um sistema multi estratégico para a recomendação e reserva de restaurantes. Para exercer essas funções o R-cube divide o processo de reserva entre três subsistemas, o sistema de recomendação de restaurantes sensível às preferências do usuário; o sistema de seleção de restaurantes que apresenta as informações necessárias para que o usuário tome uma decisão sobre o conjunto de recomendações e o sistema de reserva de restaurantes que conclui o processo. Apesar dessa diferenciação, todos os subsistemas compartilham a mesma arquitetura de múltiplas estratégias para o processamento.

Assim como em Niculescu et al. (2014), a arquitetura do R-cube apresenta as etapas de processamento de linguagem natural, controle de diálogo e geração de linguagem natural. Além disso, as entradas são pré-processadas antes do PLN. Em cada um desses níveis, múltiplas estratégias são utilizadas com o objetivo de aumentar a robustez do sistema. Os resultados de cada estratégia são sempre passados adiante, mesmo que conflitem entre si, já que os subcomponentes dos próximos níveis se beneficiam de múltiplas entradas. A arquitetura multi estratégica para um sistema de diálogo descrita nesse parágrafo está representada na Figura 3.1.

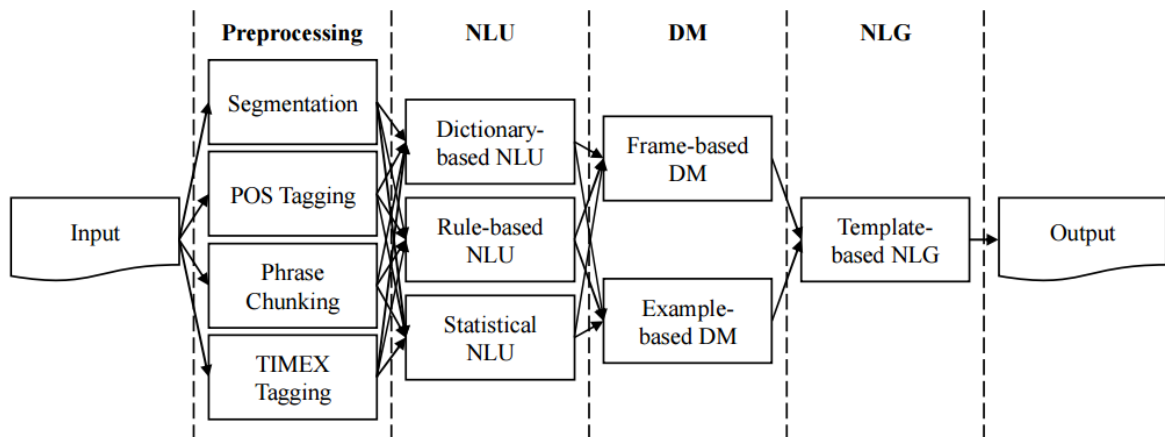


Figura 3.1: Arquitetura multi-estratégia do R-cube. Extraído de (KIM; BANCHS, 2014).

Para o pré-processamento da entrada são utilizadas quatro estratégias: segmentação, *Part-of-speech (POS) tagging*, *Phrase Chunking* e *TIMEX Tagging*. A aplicação dessas técnicas, além de normalizar a entrada, segmenta as sentenças, quando mais de uma é informada, e identifica as classes gramaticais de cada palavra, as frases preposicionais e as expressões temporais.

Durante o processamento de linguagem natural, o R-cube usa técnicas baseadas em regras e modelos estatísticos, como em D'Haro et al. (2015), mas também dicionários. O sistema usa tanto dicionários estáticos quanto dinâmicos que são construídos durante a interação do usuário com o sistema.

Para o controle do diálogo, são utilizadas a abordagem baseada em *frames* e exemplos. A abordagem baseada em *frames* utiliza os *frames* gerados na etapa anterior. Esses *frames* semânticos contém o ato de diálogo, que é a intenção da sentença que o PLN identificou e as entidades nomeadas reconhecidas. Com essas informações, o R-cube usa regras previamente construídas que definem uma resposta, o tipo de regra e um padrão de reconhecimento. A resposta está diretamente relacionada à geração de linguagem natural. O tipo define se aquela regra é uma iniciativa do agente ou se é uma requisição do usuário, identificada pelo padrão definido na regra. As regras de iniciativa conduzem o diálogo para uma situação onde o usuário responde ao agente, as regras de requisição invertem esses papéis. A abordagem baseada em exemplos para o controle do diálogo usa estruturas de índices de perguntas e respostas e tem como objetivo responder mensagens de propósito geral.

A proposta para a geração de linguagem natural é baseada em *templates* que possuem uma forma final, formas alternativas e lista de variáveis. As palavras das formas finais são apresentadas conforme o *template*. As formas alternativas e lista de variáveis são *tokens* que podem ser substituídos por palavras diferentes para gerar variabilidade. Os autores mostram um exemplo de funcionamento do R-cube, porém nenhuma avaliação do sistema foi apresentada.

A literatura de recomendação em turismo apresenta também abordagens baseadas em ontologias. Garrido et al. (2016) propõe uma solução integrada de *Embodied Conversational Agents* (ECA) e *lightweight ontologies* (LWO) para recomendação de pontos turísticos na província de Teruel na Espanha. Além disso, técnicas de inteligência artificial são utilizadas pelo ECA.

Para atingir o objetivo do trabalho, os autores utilizaram o Maxine, um motor direcionado por *scripts* para controle e visualização de ambientes ou atores virtuais. O sistema suporta interações multimodais em tempo real por canais como texto, voz e imagem e é capaz de reconhecer o estado emocional do usuário, dando suporte ao processo de decisão das ações do agente. As regras de interação dos agentes criados no Maxine são definidas utilizando *Artificial Intelligence Markup Language* (AIML).

Cada tópico de interação em AIML possui uma categoria, um padrão de entrada, um *template* de resposta e opcionalmente a última frase dita pelo agente para controle do fluxo do diálogo. Para identificar qual o tópico no contexto da conversa, métodos de busca informada são aplicados em uma árvore organizada pelos padrões de entrada dos tópicos definidos em AIML.

Para criar uma interface entre o Maxine e outros sistemas, o interpretador AIML foi modificado para que suportasse a chamada de *scripts* definidos dentro de uma nova *tag* de marcação. Essa interface permite que as respostas sejam criadas programaticamente.

A LWO criada pelos autores, chamada *Tourist Information of TERuel for Intelligent Agents* (TITERIA), agrupou as informações turísticas em 6 grandes áreas: festas, patrimônios, natureza, gastronomia, história e centros culturais. Essas categorias são divididas em subcategorias. A linguagem utilizada para a representação da TITERIA foi a *Ontology Web Language* porque permite a inserção de semântica ao *schema*. As

informações estruturadas de acordo com a TITERIA são armazenadas em um servidor que o ECA interage através de *scripts* LUA.

Para utilizar as informações do servidor, o ECA consulta sua base de dados local em AIML para determinar o *script* LUA que fará a comunicação com o servidor em que a ontologia está estruturada. O *script* obtém a informação e a decodifica para que o ECA possa entregá-la ao usuário.

Os autores descrevem os testes aplicados para a avaliação do sistema. Em um primeiro momento, o ambiente de teste foi similar a um quiosque interativo com um microfone externo e caixas de som. Porém, essa configuração apresentou condições que prejudicaram o sistema, como ruídos de fundo, diferenças de distância entre o usuário e o microfone e a captação dos sons do próprio alto-falante do ECA. Para evitar esses problemas, os autores substituíram o microfone e alto-falante por *headphones*. Os autores não apresentam no artigo quantos usuários realizaram o experimento nem qualquer tipo de avaliação do sistema feita por esses usuários.

3.2 Trabalhos anteriores da FN-Br

O foco recente do laboratório FrameNet Brasil em pesquisas relacionadas ao domínio de turismo se iniciou no domínio do esportes sob a perspectiva de turismo pelo Projeto Copa 2014. O principal produto do projeto foi o Copa 2014 FrameNet Brasil software (C-14/FN-Br), uma aplicação *web* que entrega ao usuário um dicionário trilingue baseado em *frames*, disponível em português, espanhol e inglês ⁴. Seu objetivo é explorar as vantagens da semântica de *frames* para o desenvolvimento de dicionários eletrônicos com apelo ao público geral, não apenas para especialistas da área (TORRENT et al., 2014).

A transformação da base em um recurso multilíngue para o C-14/FN-Br exigiu quatro tipos de alteração. Como todos os elementos do banco de dados existem para os três idiomas, inclusive os *frames*, uma relação de tradução teve que ser criada para mapear *frames* iguais em línguas diferentes. Nesse tipo de relação, as duas estruturas devem compartilhar todos os EFs. A segunda alteração foi a separação das sentenças anotadas entre os três idiomas. O processo de anotação de sentença marca também as funções

⁴<http://dicionariodacopa.com.br/>

gramaticais e tipos de frase encontrados em uma sentença, como essas informações variam entre idiomas, um mapeamento teve que ser criado para a comparação entre padrões de valência de ULs no dicionário. A quarta e última modificação foi a inclusão de relações entre ULs e suas traduções equivalentes.

Essas características inseridas no banco de dados são essenciais para o funcionamento dos quatro modos de navegação pelo conhecimento oferecidos pelo C-14/FN-Br. O primeiro modo não é muito diferente dos dicionários comuns: o usuário procura pela palavra que deseja saber o significado em uma lista. Quando o lema escolhido está associado a mais de um *frame*, uma tela de desambiguação é mostrada e finalmente o significado é visualizado.

Alternativamente, o usuário pode informar uma sentença inteira, nesses casos a aplicação tentará identificar qual o *frame* mais adequado para a entrada. Para isso, todos os lemas da sentença são identificados e todos os *frames* evocados pelas ULs são identificados. Como cada UL pode evocar mais de um *frame*, o sistema forma *clusters* a partir das combinações dos possíveis *frames* e de *frames* relacionados a eles. Em seguida, a distância entre os *frames* dentro de um *cluster* é medida usando pesos atribuídos a cada tipo de relação, o grupo com relações mais fortes é escolhido pelo C-14/FN-Br para determinar o significado.

Os outros dois modos de navegação estão atrelados aos *frames*, não as ULs. O usuário pode visualizar os dados de um *frame* através da navegação de uma lista organizada por domínio ou através do grafo de *frames* relacionados. Essa última forma de navegação permite que o usuário, principalmente um turista, consiga visualizar como os domínios da Copa do Mundo, Futebol e Turismo são organizados.

A análise pelos autores dos dados coletados sugere que a aplicação foi de fato utilizada por turistas: dos 1,883 acessos, 71,8% foram realizados no Brasil, e cerca de 26% de todas as sessões foram em inglês ou espanhol. Sobre os modos de consulta, a análise revela que a maioria foi realizada pelo modo de busca comum de dicionários (procurando-se a palavra), porém todos os outros modos representam parte significativa das consultas realizadas. O modo menos utilizado foi a exploração da rede de *frames* em espanhol, representando 8% das consultas nesse idioma (TORRENT et al., 2014).

Dando continuidade aos trabalhos sobre o domínio dos esportes, porém dessa vez com foco nas Olimpíadas de 2016, além de novos *frames*, as relações qualia passaram a ser incluídas na base da FrameNet Brasil (COSTA; TORRENT, 2017). A principal motivação para isso, segundo os autores, era enriquecer a base com relações entre ULs para a obtenção de melhores resultados em trabalhos de Compreensão de Linguagem Natural, Desambiguação e Tradução por Máquina.

Com suporte da ferramenta SketchEngine, foi realizada uma pesquisa em *corpora* que permitiu, através da observação de frequências e valências, identificar ULs candidatas. O domínio dos esportes foi então modelado através de 32 *frames* e 640 ULs, agregando recursos à base de dados de Turismo que contava com 52 frames e 425 unidades lexicais.

Paralelamente, o primeiro sistema de recomendação do m.knob foi proposto pela FN-Br e pelo NEnC⁵ (Núcleo de Engenharia do Conhecimento) do Departamento de Ciência da Computação da UFJF. O MKNOB-RECOMMENDER foi desenvolvido em um estudo de caso da arquitetura aberta de sistemas de recomendação MMRecommender (SILVA et al., 2017). Neste contexto, a arquitetura auxiliou principalmente na definição e construção do modelo de recomendação e na apresentação de um modelo visual de fácil entendimento.

Durante o desenvolvimento da aplicação os recursos construídos pela FN-Br para os domínios de Turismo e Esporte ainda não eram suficientes para sustentar um sistema de recomendação e por isso não foram utilizados. O sistema foi desenvolvido sobre um modelo híbrido, com filtragem baseada em conteúdo e colaborativa, que media o nível de aderência do perfil e contexto do usuário com características de locais e eventos esportivos.

Os dados de locais foram provenientes da API do Google Maps⁶ e enriquecidos semanticamente utilizando a API Babel Net⁷. As informações de eventos foram extraídas da base *Olympic Data Feed*⁸, disponibilizada de forma livre pelo comitê olímpico. Como os usuários se conectavam ao m.knob pela plataforma Facebook, o perfil de usuário era construído pelos dados da rede social. As informações de contexto utilizadas foram a busca textual e a geolocalização do usuário do sistema.

⁵<http://www.ufjf.br/nenc/>

⁶<https://cloud.google.com/maps-platform/>

⁷<http://babelnet.org/>

⁸odf.olympictech.org

Quando o usuário inicia o processo de recomendação (realizando uma busca textual ou por locais próximos), o sistema recupera o vetor de características deste usuário no banco de dados e aplica um filtro inclusivo sobre os itens conhecidos, garantindo que apenas os objetos que possuam pelo menos uma característica em comum com o usuário sejam retornados. Esse procedimento é simples e eficiente porque as características são booleanas. A tabela 3.1 apresenta alguns vetores de característica de exemplo, neste exemplo o objeto 01 seria removido quando o filtro inclusivo fosse aplicado.

Tabela 3.1: Exemplo de vetor de característica representando quatro características: Compras, *Fast-food*, Café e *Rugby* (SILVA et al., 2017).

	Compras	<i>Fast-food</i>	Café	<i>Rugby</i>
usuário	0	1	1	1
objeto 01	1	0	0	0
objeto 02	0	1	1	1
objeto 03	0	1	0	0

Após o filtro inclusivo, a similaridade de cosseno do vetor do usuário com os vetores de objetos é calculada. Quando a geolocalização do usuário é conhecida, a distância euclidiana em relação a cada atração também é computada. As duas medidas são normalizadas através da divisão pelo maior valor de cada métrica. A matriz de decisão final é construída através do ponderamento dos valores normalizados: 0,7 para a medida de similaridade de cosseno e 0,3 para a distância euclidiana (que também é utilizada para o desempate). O ranqueamento é feito sobre os valores dessa matriz.

O sistema de recomendação foi avaliado em duas etapas, na primeira a eficácia do sistema foi aferida através de um experimento controlado em que foram definidos perfis fictícios de usuário e os itens considerados mais aderentes a esses perfis. As métricas *Precision*, *Recall*, *F1 Score* e *G measure* foram calculadas e foram obtidos os resultados 96,552%, 98,276%, 97,406% e 97,410%. A segunda etapa consistiu da observação das ações de usuários aleatórios do sistema e foram medidos os números de compartilhamento e *likes* após a recomendação de locais pelo sistema.

Em (COSTA et al., 2018), o processo de modelagem dos domínios de Turismo e Esportes para o m.knob são descritos e é proposto um processo para a categorização de locais turísticos baseada na informação semântica de comentários de plataformas online.

O processo de modelagem da base incorporou 250 *frames* da FrameNet Berkeley

a base do m.knob. Para a escolha desses *frames*, foram considerados 3,495 comentários em inglês sobre 939 locais turísticos de São Francisco. Primeiramente, ULs candidatas foram extraídas do *corpus* usando a ferramenta TermoStat, em seguida, os *frames* relacionados foram ranqueados por frequência. Por fim a relevância dos *frames* mais frequentes para o domínio foi avaliada por linguísticas.

Além das relações qualia, os autores também descrevem a incorporação de um outro conjunto de relações para representar os casos em que participantes de um *frame* podem ser definidos por outro *frame* ou relações metonímicas. Para exemplificar a contribuição desse conjunto para a estruturação do conhecimento, os autores mostram como o *frame* *Attracting_tourist*, pode ser evocado pela sentença "The city lures Brazilians with beautiful beaches and nice shops"⁹. Os EFs do *frame* seriam relacionados aos *frames* *Locale*, *People*, já que eles definem respectivamente o local da atração e o turista. Além disso, o adjetivo brasileiro (*Brazilian.a*) e o substantivo pessoa devem ser conectados (*people.n*). Com essas relações, as ULs da sentença, através da rede estão conectadas a *Attracting_tourist* com poucos arcos.

A categorização dos locais turísticos é baseada nas avaliações subjetivas de usuários presentes nos comentários em plataformas online. O categorizador extrai desses comentários representações semânticas das atrações a fim de resolver o problema de sobrecarga de informação presente nessas plataformas. Para isso, o sistema identifica os *frames* evocados em todos os comentários de determinado local e atribui um peso a cada UL e cada *frame* para uma atração utilizando a função logística:

$$w = \frac{1 - e^{ca}}{1 + e^{ca}}$$

Onde c é o número de ocorrências da UL ou *frame* e a é o peso médio da ativação da UL ou *frame* nos comentários da atração. Os pesos são usados na conexão da atração com os *frames* relacionados. Além dessas conexões são armazenadas informações do local como nome, localização, horário de funcionamento e classificação segundo a plataforma online.

Para recomendar locais o sistema primeiro identifica candidatos a UL na entrada

⁹A cidade atrai brasileiros com belas praias e boas lojas (Tradução nossa).

(usando *bigrams*) e desambigua os lemas que apontam para mais de uma UL, usando as relações criadas na base e as distâncias na rede. Depois, é gerado um *cluster* que contém as ULs extraídas, palavras da entrada encontradas nos comentários dos locais e ULs relacionadas aos *frames* evocados pelas ULs extraídas. Esse *cluster* é então usado para iniciar um procedimento de *spreading activation*, descrito em (MATOS et al., 2017), sobre o grafo formado por atrações ligadas a *frames*, ULs e palavras dos comentários do local. Os locais com maior ativação são recomendados.

4 Arquitetura do m.knob

Neste Capítulo os componentes principais da arquitetura do m.knob serão apresentados em detalhe. Esses componentes, as tecnologias usadas para seu desenvolvimento e suas interações estão esquematizadas na Figura 4.1.

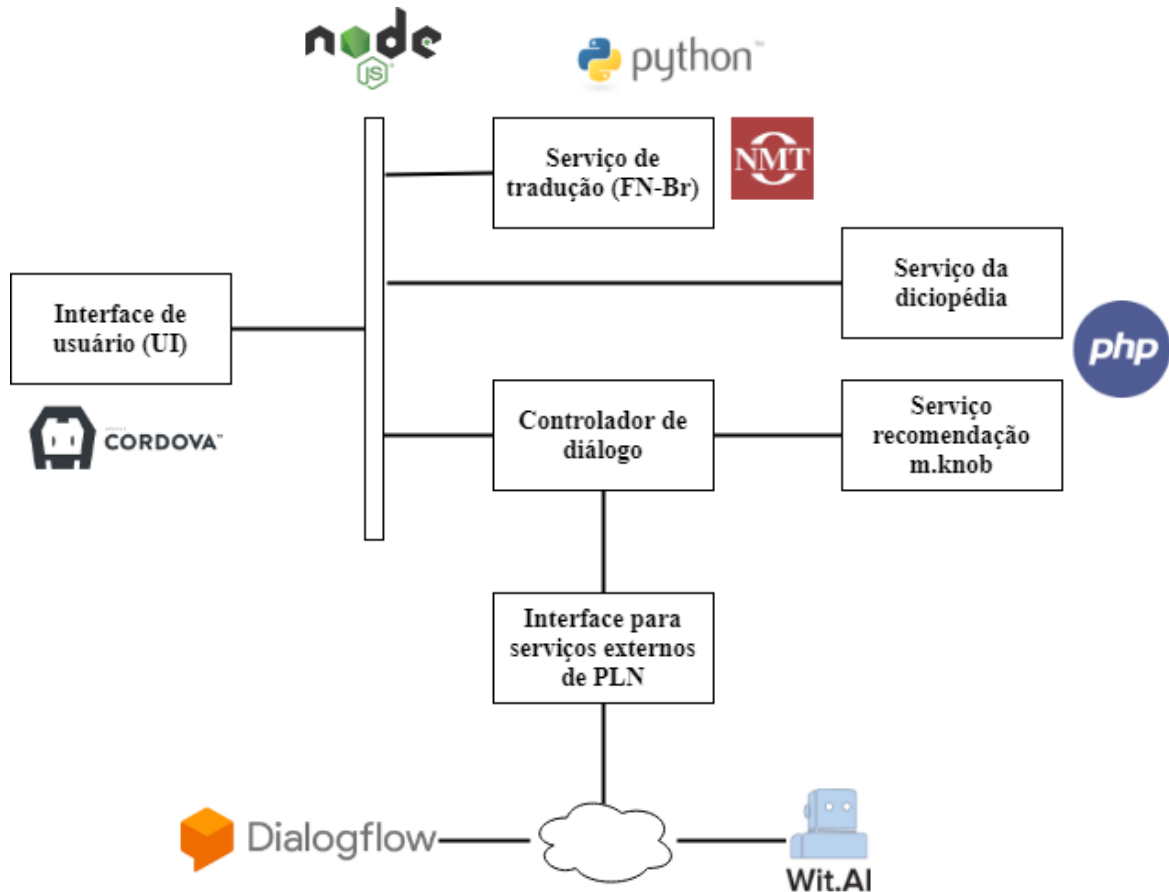
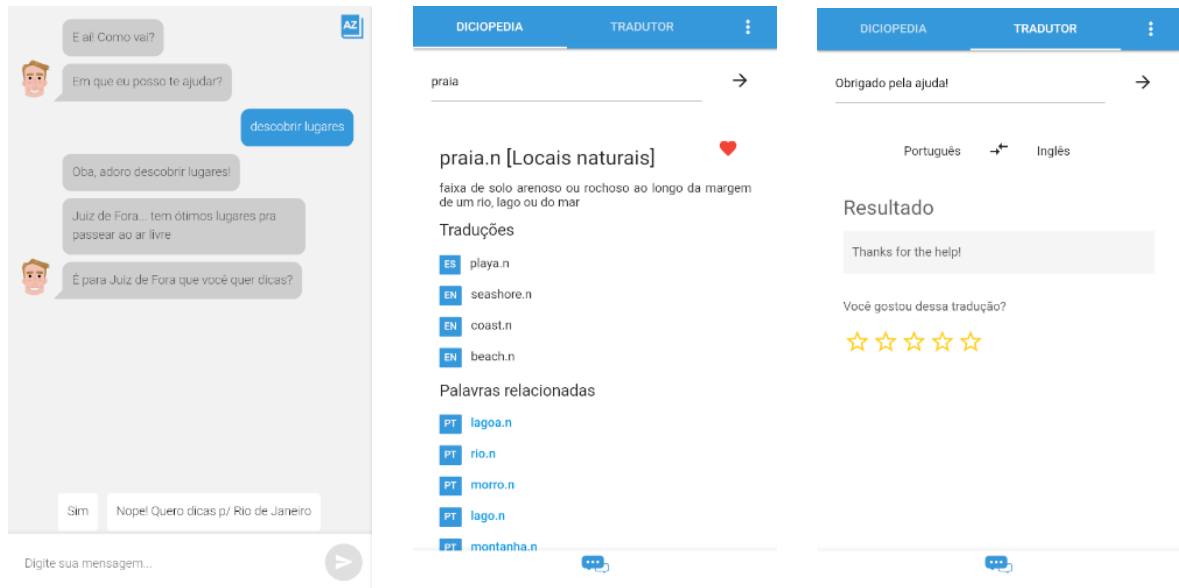


Figura 4.1: Componentes da arquitetura do m.knob.

4.1 Interface de usuário

O usuário interage com o sistema através de uma interface gráfica de usuário (GUI) em uma aplicação para dispositivos móveis Android ou iOS, desenvolvida em JavaScript utilizando o *framework* Apache Cordova¹⁰. O aplicativo pode ser utilizado em português, inglês ou espanhol e possui três telas distintas para cada uma de suas funcionalidades

¹⁰<https://cordova.apache.org/>

(a) Bate-papo com o *chatbot*.

(b) Visualização da palavra “praia” na dicionária.

(c) Tradutor do m.knob.

Figura 4.2: Telas do aplicativo em português.

principais: um *chat* para que o usuário possa interagir com o *bot*, uma tela de tradução e uma de navegação na dicionária, uma enciclopédia multilíngue que conecta palavras relacionadas. A Figura 4.2 apresenta as três telas principais do aplicativo em português.

Tanto a dicionária quanto o tradutor são interfaces secundárias, isto é, por padrão o aplicativo sempre é aberto no *chat* com o *bot*. O usuário pode então clicar em um botão no topo da tela ou deslizar para a esquerda para acessar essas duas funcionalidades (Figura 4.2a). Na aba da dicionária, existe um campo para que o usuário possa buscar por termos. O resultado dessa busca é exibido em uma lista de *links* que o usuário pode clicar e abrir a visualização detalhada do termo, como na Figura 4.2b. Já nessa visualização, são apresentados também *links* de termos relacionados que permitem que o usuário navegue como desejar na dicionária.

A tela do tradutor possui um campo de entrada e um de resultado, além das estrelas utilizadas para a avaliação da tradução do sistema. O usuário pode ainda escolher qual o idioma da entrada e o da saída, nesse momento porém, somente as traduções de português para inglês e vice-versa podem ser utilizadas. Todos esses elementos são apresentados na Figura 4.2c. Para traduzir as sentenças a aplicação móvel interage com o servidor NodeJS do m.knob que age como intermediário na comunicação com o serviço em Python do tradutor.

A interface de bate-papo é a visão principal do sistema porque é através dela que o usuário interage com o *chatbot*, buscando por atrações de interesse e visualizando os resultados dessa busca. Essa parte do aplicativo é responsável por ler as entradas do usuário, enviá-las para o gerenciador de diálogo e criar a visualização das saídas do gerenciador de diálogos para o usuário.

As saídas do gerenciador de diálogo são conjuntos de ações que a interface deve realizar e sugestões de resposta rápida que serão exibidas para o usuário de acordo com o contexto da conversa. As ações a serem realizadas pela interface podem ser classificadas em três tipos: exibir uma fala do *chatbot*, apresentar *cards* de atrações ou abrir outra tela do aplicativo (dicionário ou tradutor). Na Figura 4.2a, além de balões de conversa, as opções rápidas de resposta são mostradas para o usuário.

4.2 Interações com serviços externos

O aumento recente do interesse por *chatbots* impulsionou o desenvolvimento de serviços para o desenvolvimento de *bots*. Dentre os mais populares destacam-se o Wit.ai¹¹ e o Dialogflow¹² do Facebook e da Google, respectivamente. Ambos são usados no m.knob para facilitar a construção de diálogos.

Como parte considerável da rede do m.knob foi construída para suportar a funcionalidade de recomendação, não foram propostos métodos para o processamento de outros tipos de requisição, como por exemplo, saber pela sentença que o usuário gostaria que outra tela do aplicativo fosse aberta. Para esse tipo de requisição geral, foi desenvolvido um componente. Ferramentas como Wit.ai e Dialogflow são projetadas para esse tipo de funcionamento e realizam bem essas tarefas quando possuem um bom número de sentenças de exemplo (conjunto de treinamento).

O componente local que se comunica com as APIs *web* desses serviços abstrai o formato das requisições e padroniza as respostas, permitindo que o controlador de diálogo consuma tanto o Wit.ai quanto o Dialogflow através de uma única interface.

¹¹<https://wit.ai/>

¹²<https://dialogflow.com/>

4.3 Controlador de diálogo

O controlador de diálogo é um componente essencial para qualquer *chatbot*. Sua responsabilidade é gerenciar o contexto da conversa e resolver qual resposta será utilizada, dado processamento de uma entrada do usuário. Geralmente, as respostas são baseadas em *templates* porque essa é uma forma fácil de se gerar linguagem natural, apesar da falta de variedade nas respostas. Algumas trocas simples de palavras nos *templates* no entanto, podem ajudar a mitigar esse problema (NICULESCU et al., 2014). Dadas as dificuldades em se gerar linguagem natural a partir de significados, esse trabalho também segue a abordagem de *templates* de resposta.

Administrar o contexto de uma conversa também é um desafio no desenvolvimento de *chatbots* e um problema conhecido por pessoas que já interagiram com esse tipo de agente. Para evitar esses problemas alguns *chatbots* são desenvolvidos “sobre trilhos”, isto é, seguem um roteiro de conversa pré-definido que é um fluxo sobre estados e transições. A escolha de qual abordagem seguir ao projetar um *chatbot* deve ser orientada à tarefa que ele irá cumprir para o usuário. Tome por exemplo um *chatbot* que interage com o usuário para realizar pedidos de pizza. Para realizar sua função o agente deve saber qual o tamanho, o sabor da pizza, o tipo de borda e a bebida de acompanhamento. Utilizar um roteiro de conversa fixo nesse caso ajuda na obtenção dessas informações e na manutenção do estado da conversa, o que é importante para saber quais informações ainda devem ser solicitadas ao usuário.

No caso do m.knob, o objetivo principal da aplicação é oferecer ao usuário um assistente pessoal capaz de fazer boas recomendações de locais turísticos. Para recomendar as atrações, a interface captura a entrada do usuário. Porém, esse não é o único dado importante para o sistema de recomendação. Atualmente, também é necessário conhecer qual é a cidade dos locais que serão sugeridos e caso esse trabalho seja expandido no futuro, mais parâmetros podem se tornar necessários. Para essa etapa de configuração onde o usuário informa esse tipo de dado, a definição de um roteiro da conversa é benéfica.

Considerados esses fatores, o controlador de diálogo foi desenvolvido como uma *bot engine* simplificada, onde um fluxo de conversa é definido através de arquivos de configuração e o componente é capaz de navegar por seus elementos utilizando as entradas

```
1  {
2    "ending":{
3      "pt":[
4        "Tchau!",
5        "Até mais!",
6        "Nos falamos mais tarde!"
7      ],
8      "en":[
9        "Bye!",
10       "See ya later!",
11       "Talk to you later!",
12       "See you in a bit!",
13       "Make sure you come back, and we'll have another chat!"
14     ],
15     "es":[
16       "¡Adiós!",
17       "¡Hasta luego!",
18       "¡Hasta ahora!"
19     ]
20   },
21   "fallback":{
22     "pt":[
23       "Minhas mais sinceras desculpas. Eu sou apenas um robô, não tenho todas as respostas"
24     ],
25     "en":[
26       "My most humble apologies. I am but a modest machine, I don't know everything"
27     ],
28     "es":[
29       "Mis más sinceras disculpas. Yo soy sólo un robot, no tengo todas las respuestas"
30     ]
31   ]
32 }
```

Figura 4.3: Exemplo de definição das falas do *chatbot*.

recebidas da interface. Apesar de permitir a definição de roteiros pouco flexíveis, o controlador que foi desenvolvido para o *m.knob* é capaz de delegar a navegação pela conversa a funções definidas pelo programador. Dessa maneira, podem ser definidas ações complexas (inclusive assíncronas) no processamento das entradas do usuário que permitem que o *chatbot* funcione fora de um roteiro rigidamente definido.

Para o funcionamento do *chatbot*, o controlador deve verificar dois arquivos JSON de configuração. O primeiro armazena as falas do *bot*. Cada fala possui uma chave única que apresenta uma ou mais variações da mesma fala em cada um dos idiomas suportados. A Figura 4.3 exemplifica como é feito esse armazenamento. Como os arquivos de fala são

```

27   "2":{
28     "quickreplies":null,
29     "events":{
30       "onenter":"action:locationDefinition",
31       "onexit":"action:saveLocation"
32     }
33   },
34   "3":{
35     "quickreplies":null,
36     "freeText":{
37       "handler":"action:processText"
38     }
39   },
40   "4":{
41     "quickreplies":null,
42     "freeText":{
43       "handler":"action:processText"
44     }
45   },
46   "5":{
47     "quickreplies":[
48       {
49         "option":"great",
50         "transition":"J"
51       },
52       {
53         "option":"notWhatIWant",
54         "transition":"I"
55       }
56     ]
57   },
277  "F":{
278    "actions":[
279      "smalltalk:skip",
280      "smalltalk:whatsInYourHead"
281    ],
282    "targetState":"3"
283  },
284  "G":{
285    "actions":[
286      "smalltalk:askHint"
287    ],
288    "targetState":"4"
289  },
290  "I":{
291    "actions":[
292      "smalltalk:fallback",
293      "smalltalk:tryAgain"
294    ],
295    "targetState":"6"
296  },
297  "J":{
298    "actions":[
299      "action:matchBasedTalk",
300      "smalltalk:onIt",
301      "action:showCards",
302      "smalltalk:askFeedback"
303    ],
304    "targetState":"7"
305  },
306  "K":{
307    "actions":[

```

(a) Configuração de alguns estados.

(b) Configuração de algumas transições.

Figura 4.4: Exemplo de configuração de roteiro do *chatbot*.

pequenos, seus dados são mantidos na memória principal do servidor.

No outro arquivo de configuração está a definição do fluxo de conversa que deve ser seguido. Nesse documento JSON existem cinco chaves principais: uma para a definição dos estados, outra para transições, gatilhos de transição, transições iniciais e opções de respostas rápidas. As opções de resposta rápida apenas definem os textos que serão mostrados para o usuário quando essas opções forem exibidas, enquanto a chave de transições iniciais armazena um subconjunto das transições que pode iniciar o fluxo.

As definições de estados armazenam as opções que serão exibidas para o usuário nesse estado e as transições que devem ser aplicadas caso elas sejam escolhidas. Quando um estado permite que o usuário digite uma sentença como resposta, a função responsável por tratar essa resposta deve ser indicada. Por fim, na definição também podem constar ações a serem executadas ao entrar ou sair do estado. Alguns exemplos de estados são apresentados na Figura 4.4a.

Nos fluxos do m.knob, um estado representa um momento da conversa em que o sistema aguarda uma resposta do usuário enquanto as transições representam as ações do *bot*. Naturalmente, as transições possuem um estado de destino e uma lista de ações do *chatbot* que serão exibidas ao usuário pela interface. As ações podem ser do tipo *smalltalk* ou *action*, representando respectivamente uma fala do agente ou a execução de um método do programa (a exibição das atrações é feita através de uma *action*). A Figura 4.4b mostra a configuração de algumas transições do *chatbot*.

A última configuração é a dos gatilhos de transição, que são identificados por uma chave e se relacionam a uma transição. A ideia é que essas chaves representem algum tipo de retorno de outros serviços que se comuniquem com o controlador de diálogos. A configuração foi desenvolvida para aplicar transições baseando-se nos *intents* identificados no PLN do Wit.ai ou do Dialogflow. Nesse caso, os *intents* são as próprias chaves. Na configuração atual, por exemplo, existe uma chave chamada "abrir_tradutor" que identifica uma transição que deve ser aplicada caso um serviço externo retorne esse *intent*.

5 Sistema de recomendação

O modelo de recomendação proposto neste trabalho para o m.knob utiliza a rede de atrações turísticas, palavras, ULs e *frames* como base de conhecimento onde as *queries* de consulta são formadas a partir das próprias sentenças dos usuários durante as conversas com o *chatbot*. Todas as buscas realizadas na base estão incluídas dentro de uma sessão que identifica uma interação entre o usuário e o *chatbot*.

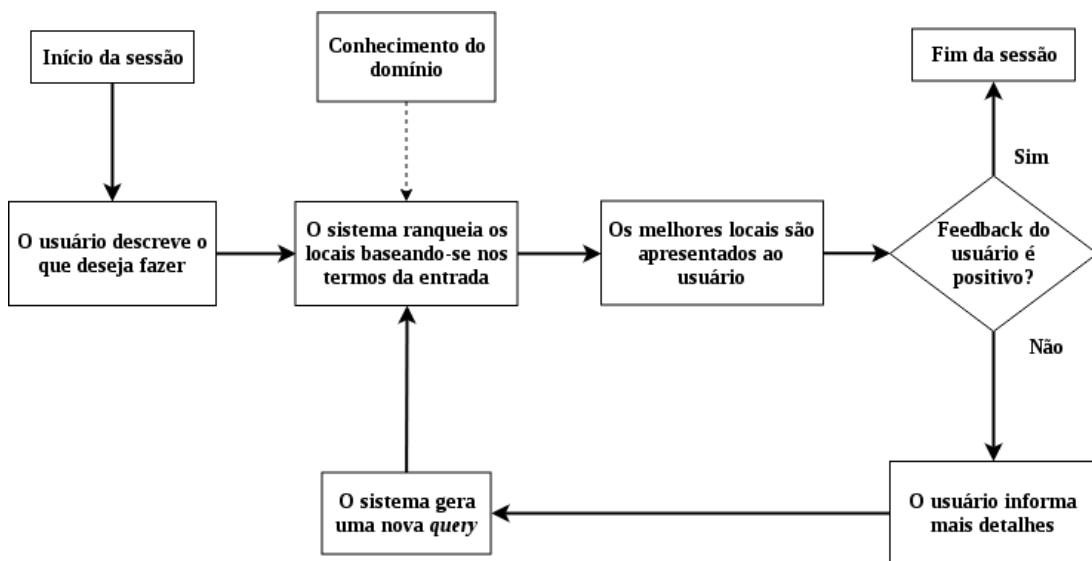


Figura 5.1: Ciclo de *feedback* do sistema de recomendação do m.knob.

Esse agrupamento das *queries* em sessões é uma característica comum de sistemas de recomendação baseados em conhecimento e, neste trabalho, permite que o usuário refine sua busca através de críticas. Essas críticas são uma forma de se obter mais termos (a partir de entradas do usuário em texto livre) para a recomendação e podem tanto restringir o número de itens sendo recomendados quanto incluir novos itens quando as informações conhecidas anteriormente não eram suficientes. A Figura 5.1 apresenta o ciclo de *feedback* do sistema de recomendação do m.knob. Uma sessão inicia-se quando o usuário pede recomendações ao *bot*, toda entrada feita como crítica é parte dessa sessão. É o controlador de diálogo, apresentado na seção 4.3, que informa se o usuário está criticando as recomendações da sessão. A sessão termina quando o usuário avalia positivamente as respostas do *chatbot* ou quando decide começar novamente a interação, possivelmente

porque as sugestões recebidas foram insatisfatórias.

No momento da escrita deste trabalho, as recomendações estão limitadas a locais das cidades de Juiz de Fora e Rio de Janeiro. Os comentários e informações de atrações dessas cidades foram coletados para a condução dos primeiros testes da aplicação e espera-se que a metodologia utilizada para a obtenção desses dados possa ser utilizada na inclusão futura de atrações de outras cidades.

5.1 A rede do m.knob

O m.knob (Multilingual Knowledge Base) é uma aplicação onde as principais funcionalidades são construídas sobre uma base de conhecimento, como sugerido pelo nome. Essa dependência destaca a importância da modelagem do domínio para o funcionamento adequado do sistema. A construção do modelo, realizada pelos linguistas da FrameNet Brasil, foi feita iterativamente sempre agregando dados dos modelos de trabalhos anteriores, como descrito na Seção 3.2. Nesta seção, o processo de construção da rede utilizando sistemas de recomendação é relatado, assim como a estrutura final é exposta.

5.1.1 Categorização de atrações

Para ser capaz de recomendar locais, o sistema deve conhecer seus atributos. Assim como em (COSTA et al., 2018), neste trabalho os locais foram caracterizados utilizando avaliações de usuários em comentários de plataformas online, porém, além dos *frames* extraídos dos comentários, foram utilizados n -grams ($n \geq 2$) e lexemas. Foram considerados os lexemas presentes em comentários e no texto de descrição do local.

A extração de n -grams foi feita sobre os comentários com palavras anotadas (*POS tagging*), considerando aqueles que se encaixavam em padrões definidos pelos linguistas da FN-Br. Os mais frequentes dentre os comentários são associados ao local referente e as palavras que o compõe.

5.1.2 Incorporação de Frames aos papéis Qualia

A integração de relações qualia a uma FrameNet torna a rede mais densa e estabelece conexões entre ULs, efetivamente aumentando a capacidade da rede em resolver problemas de desambiguação e compreensão de linguagem natural. Essas relações porém, são pouco específicas. Para resolver esse problema, um relação ternária foi acrescentada à rede do m.knob.

A relação ternária especifica uma relação qualia intermediada por um *frame*, onde as ULs são associadas a EFs. A modelagem dessas relações exige que o *frame* seja o mais genérico possível sem que conflite com o quale ou o generalize demais. Os *frames* e os EFs utilizados (com nomes traduzidos) nas relações ternárias do m.knob estão descritos na Tabela 5.1.

Tabela 5.1: Relações qualia ternárias na base do m.knob.

Quale	Frame	EF/UL1	Especificação	EF/UL2
Agentivo	Afetar_intencionalmente	Agente	efeito_agentivo	Paciente
Agentivo	Causacao	Ator	causa_agentiva	Efeito
Agentivo	Agir_intencionalmente	Agente	causa_agentiva	Acao
Agentivo	Causacao	Efeito	causado_por	Causa
Agentivo	Criacao_culinaria	Comida_produzida	criado_por	Cozinheiro
Agentivo	Criar_intencionalmente	Entidade_criada	criado_por	Criador
Agentivo	Atingir_primeiro	Nova_ideia	criado_por	Conhecedor
Constitutivo	Influencia_objetivo	Entidade_influenciante	afeta	Entidade_dependente
Constitutivo	Pessoas_por_origem	Person	tem_origem	Origem
Constitutivo	Pessoas_por_religiao	Pessoa	e_um_seguidor_de	Religiao
Constitutivo	Agir_intencionalmente	Agir	e_atividade_constitutiva_de	Agente
Constitutivo	Atributos	Atributo	e_atributo_constitutivo_de	Entidade
Constitutivo	Contendo	Conteudo	esta_em	Recipiente
Constitutivo	Inclusao	Parte	esta_incluida_em	Total
Constitutivo	Ingredientes	Produto	efeito_de	Material
Constitutivo	Associacao	Membro	e_membro_de	Grupo
Constitutivo	Parte_interior_exterior	Parte	e_parte_de	Todo
Constitutivo	Parte_todo	Parte	e_parte_de	Todo
Constitutivo	Parte_pedaco	Pedaco	e_parte_de	Substancia
Constitutivo	Construindo_subpartes	Parte	e_parte_de_construcao	Todo
Constitutivo	Parentesco	Ego	parentesco	Alter
Constitutivo	Residencia	Residente	vive_em	Local
Constitutivo	Criador	Criador	produz_naturalmente	Entidade_criada
Formal	Exemplar	Instancia	instancia_de	Tipo
Formal	Tipo	Subtipo	tipo_de	Categoria
Télico	Aptidao	Evento	e_a_habilidade_de	Entidade
Télico	Agir_intencionalmente	Agir	e_a_atividade_de	Agente
Télico	Costume	Comportamento	e_o_habito_de	Protagonista
Télico	Vício	Vício	e_o_habito_de	Viciador
Télico	Proposito_ferramenta	Ferramenta	destinado_a	Proposito
Télico	Criar_intencionalmente	Entidade_criada	objeto_da_atividade	Criador
Télico	Usando_recurso	Recurso	usado_por	Agente

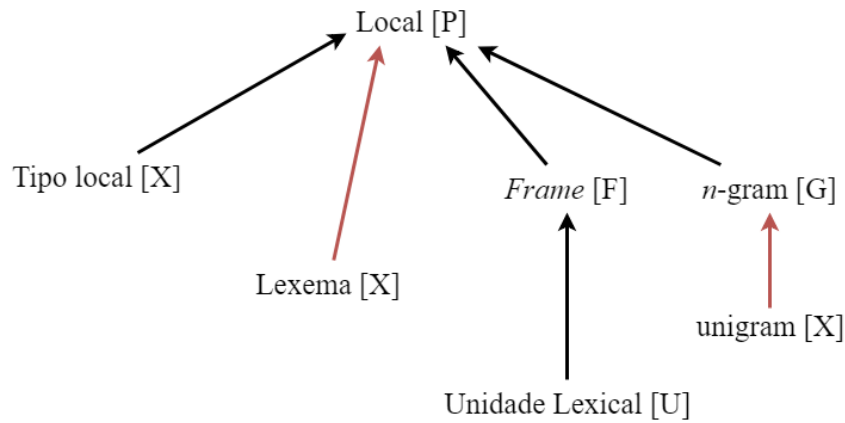


Figura 5.2: Relações entre nós do grafo do m.knob. Arcos em vermelho possuem peso 0,5.

5.1.3 Pesos das relações

O processo de ativação da rede utilizado para o ranqueamento dos locais de recomendação depende da definição de pesos para as relações do grafo. No grafo do m.knob, o conjunto de nós é constituído de cinco tipos de nós diferentes: P , G , X , F e U , que denotam locais, n -grams, lexemas, *frames* e ULs, respectivamente. As relações (G, P) , (U, F) , (F, P) têm peso igual a 1, enquanto (X, G) tem sempre peso 0,5. Para as associações entre lexemas e locais, o peso 1 é dado para aquelas em que o lexema descreve o tipo de local, o restante recebe peso 0,5. Os pesos foram definidos dessa maneira para que as relações de *frames* e *n-grams* pudessem contribuir de maneira mais significativa para a ativação dos locais, já que representam estruturas dos domínios da aplicação, enquanto os lexemas não estão sob a mesma restrição.

A Figura 5.2 mostra como os nós se conectam no grafo do m.knob. Note, pela direção dos arcos, que a rede é um grafo acíclico dirigido (DAG) e os nós de locais são os sumidouros, isto é, possuem um grau de saída nulo. A ausência de ciclos torna o procedimento de ativação discutido na Seção 5.2 mais simples e eficiente, enquanto a existência de sumidouros apenas do tipo P garante que para qualquer ativação o procedimento sempre chega a uma atração turística que pode ser recomendada.

5.2 Ranqueamento de atrações

O algoritmo utilizado nesse trabalho para a identificação de atrações turísticas baseia-se em métodos de *spreading activation*, que são utilizados para buscas em redes (MATOS et al., 2017; COSTA et al., 2018). De maneira geral o método consiste em atribuir valores de ativação a um conjunto inicial de nós que espalharão essa "energia" pela rede através de suas conexões. A ativação de novos nós implica no decaimento da "energia", dessa maneira nós distantes do conjunto inicial tendem a ter valores de ativação mais baixos, a não ser que sejam ativados por várias origens diferentes.

No algoritmo do m.knob, a definição do conjunto de ativação inicial da rede é feita em duas etapas: primeiramente, são identificados os lexemas da sentença do usuário. A partir desses, os lexemas relacionados através de papéis qualia também são recuperados. A união de ambos conjuntos consiste nos nós (X ou quando conectados a um *frame*, U) da rede que iniciarão o procedimento, a energia inicial de ativação é 1 para lexemas da sentença e 0,9 para aqueles trazidos através de uma relação qualia.

O valor de ativação para os outros nós da rede é:

$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} = \sum_i w_{ik} x_i \quad (5.1)$$

Onde \mathbf{x} é o vetor de valores de saída dos nós da rede que incidem em k e \mathbf{w}_k é o vetor de pesos dos arcos incidentes. Estes pesos estão apresentados na Figura 5.2. O valor de saída de cada nó é definido pela seguinte função logística da ativação:

$$o_k(a_k(\mathbf{x})) = \frac{1 - e^{-5a_k(\mathbf{x})}}{1 + e^{-2a_k(\mathbf{x})}} \quad (5.2)$$

A curva da Equação 5.2 em função do valor de ativação é apresentada na Figura 5.3.

Conhecidas as funções de ativação e saída de cada nó da rede, o algoritmo pode ser iniciado a partir de um conjunto de lexemas até que se conheça o valor de ativação dos nós que representam atrações turísticas. Pela própria estrutura da rede e pelo fato de ser acíclica, o algoritmo termina quando todos os caminhos alcançarem um nó P e isso ocorre após a ativação de no máximo, três níveis. Note que nem todos os locais são ativados,

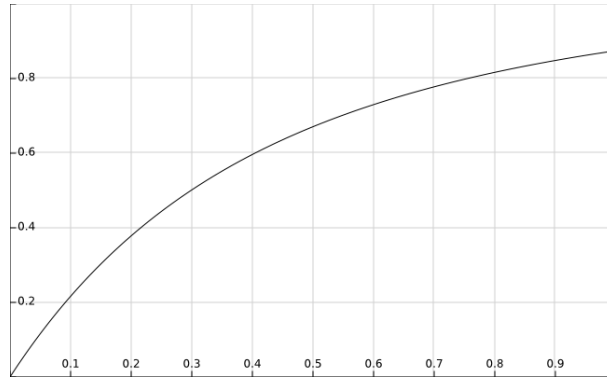


Figura 5.3: Curva da Equação 5.2.

pois a ativação pode ter sido iniciada apenas por termos que não estejam relacionados, direta ou indireta, a uma determinada atração. O algoritmo, porém, sempre encontra pelo menos uma atração, desde que exista um lexema conhecido na entrada do usuário.

Ao fim do procedimento descrito, o valor de ativação dos locais é utilizado para o ranqueamento das atrações. Em caso de empate o sistema ainda não utiliza um segundo critério e coloca a frente o primeiro local encontrado na lista original. O procedimento de ativação é apresentado no Algoritmo 1.

Algoritmo 1: Ativação de locais

```

1 lexemas ← Lexemas(entrada)
2 filaAtivação ← lexemas ∪ LexemasQualia(lexemas)
3 while filaAtivação ≠ ∅ do
4   nó ← Primeiro(filaAtivação)
5   Remove(nó, filaAtivação)
6   saída ← CalcularSaída(nó)
7   for arco ∈ Conexões(nó) do
8     Ativar(arco, saída)
9     Adicionar(arco.alvo, filaAtivação)
10  end for
11 end while

```

É importante destacar também que os arcos de nós F para nós P, isto é, as relações entre *frames* e atrações só são consideradas no procedimento quando a entrada do usuário não contém lexemas conectados a uma atração ou a um *n*-gram. A explicação para isso está no fato de que duas ULs que evocam o mesmo *frame* têm significado diferentes, porém, ainda estão relacionadas através de um certo conhecimento. Quando o sistema tenta identificar *frames* evocados pela entrada do usuário para encontrar atrações, ele está

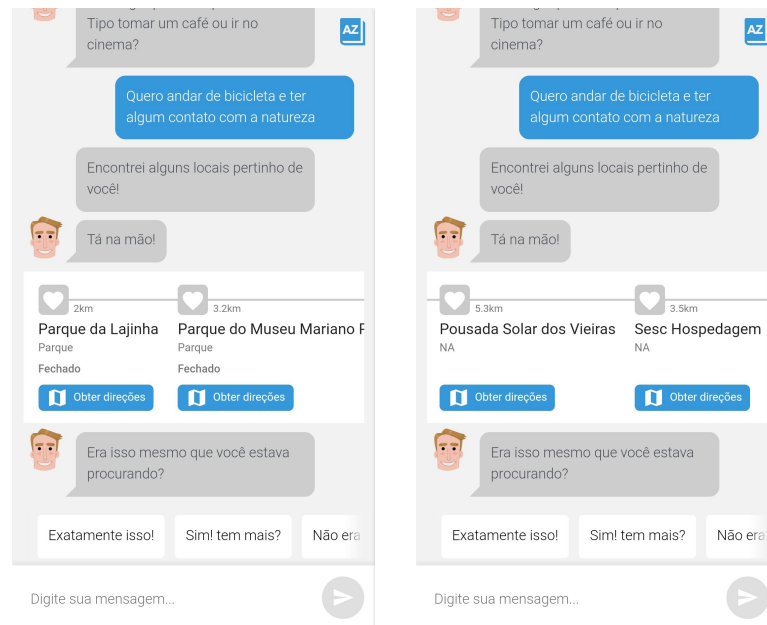


Figura 5.4: Interação bem sucedida com o sistema de recomendação através do *chatbot*.

recorrendo a essas relações mais fracas para conseguir ter algum tipo de retorno.

Note que os *frames* não são necessários quando os lexemas estão ligados diretamente à atração, porque no momento em que os comentários do local foram processados as palavras que evocaram o *frame* também foram associadas ao local. Logo, utilizar os *frames* apenas acrescentaria mais um passo de ativação e aumentaria as chances de que outros locais fossem ativados porque o mesmo *frame* foi evocado por uma UL diferente.

Ao final do procedimento completo a lista de atrações ordenada por ativação ainda é filtrada para garantir que sejam retornadas apenas atrações localizadas na cidade especificada pelo usuário. A Figura 5.4 apresenta uma interação que exemplifica a utilização do sistema de recomendação pela interface conversacional.

5.3 Adaptação às críticas do usuário

Ao ranquear locais em relação a uma entrada o objetivo do sistema é encontrar a ordenação mais próxima àquela que o usuário definiria. Erros são esperados nesse processo porque as características de um local podem ser subjetivas, ausentes no banco de dados ou as *queries* do usuário podem ser pouco específicas.

Um dos motivos pelos quais as entradas podem ser pouco específicas é a dificul-

dade em atribuir valor às características do domínio. Nesses casos, exibir uma lista de recomendação ajuda o usuário a perceber quais são os atributos desejados (AGGARWAL, 2016). A partir desse momento, para o refinamento da recomendação, o sistema deve permitir ao usuário criticar os resultados apresentados anteriormente.

Essas ações constituem o ciclo de *feedback* da Figura 5.1. A interface conversacional do sistema possibilita o uso de críticas compostas de maneira orgânica, porque uma sentença do usuário pode implicar na mudança de vários atributos na filtragem de locais de interesse. Esse tipo de crítica permite uma navegação mais rápida pelo espaço de atributos e locais, potencialmente reduzindo o número de ciclos até que o usuário aceite uma recomendação.

Para realizar um novo ranqueamento, o sistema inclui todas as palavras da crítica do usuário ao conjunto de entradas da sessão, que é constituído dos lemas da primeira *query* do usuário e das críticas realizadas no ciclo de *feedback*. O procedimento descrito na Seção 5.2 é executado novamente para a obtenção de um novo *ranking* que considere a nova crítica. Sob a premissa na qual a percepção do usuário sobre as características que ele procura aumenta com o número de recomendações recebidas, o sistema deve atribuir um peso maior para as informações da última crítica. Na prática, os lexemas dessa crítica serão ativados com mais energia na execução do algoritmo de *spreading activation*, aumentando também as chances de locais diferentes serem posicionados no topo do *ranking*. O sistema apresenta então uma nova lista de atrações priorizando os maiores valores de ativação.

Uma abordagem que também foi considerada para o endereçamento das críticas consiste em reordenar sucessivamente a primeira lista de recomendações a cada nova entrada. Para isso, os atributos e suas contribuições ao *score* de ativação de cada item da lista deveriam ser armazenados durante a sessão. Os atributos identificados em cada crítica teriam seus pesos aumentados e os *scores* dos locais relacionados atualizados para refletir essa mudança. O principal problema desse tipo de abordagem está em escalar o sistema: o aumento do número de usuários simultâneos resultaria em um número maior de listas a serem mantidas; similarmente, uma base maior implicaria em listas cada vez maiores que constantemente teriam que ser atualizadas. Observe que nenhum desses

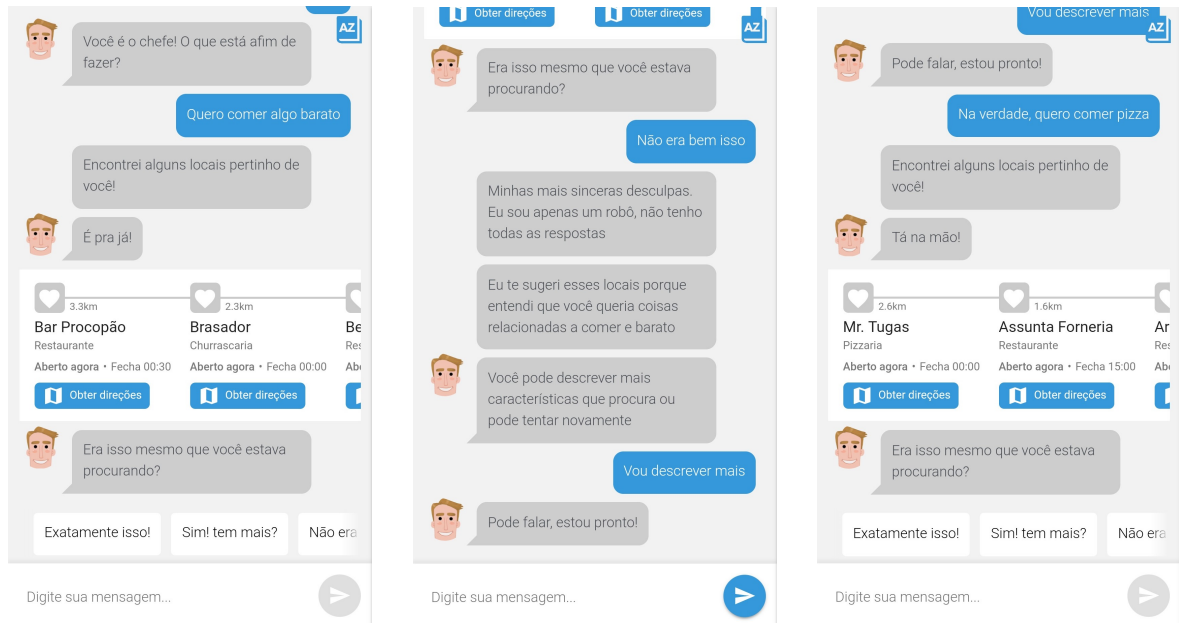


Figura 5.5: Exemplo de ciclo de *feedback* do sistema de recomendação.

problemas está presente no primeiro procedimento porque os únicos dados de uma sessão são os textos do próprio usuário.

A Figura 5.5 exemplifica uma interação onde o usuário opta por informar uma segunda sentença a fim de obter uma recomendação melhor. Note que inicialmente o sistema recomenda um restaurante e uma churrascaria, mas depois da sentença "Na verdade quero comer piza" recomenda uma pizzaria e um restaurante que também serve pizzas. É importante observar também que o *chatbot* informa ao usuário quais palavras foram consideradas pelo algoritmo de ativação, dessa maneira o usuário pode entender quais lexemas da sua sentença foram considerados para o levantamento das atrações. Essas explicações do *chatbot* podem ser expandidas em um trabalho futuro para informar, além dos lexemas reconhecidos, as relações entre eles. Dessa maneira, o procedimento de recomendação pode ser ainda mais transparente para o usuário.

6 Considerações finais

O presente trabalho apresentou um assistente pessoal de viagem capaz de recomendar atrações turísticas a partir da entrada dos usuários. A arquitetura desenvolvida permite que futuramente outros *chatbots* possam ser desenvolvidos de maneira bem similar para outros domínios e utilizando outros modelos. A metodologia utilizada para a classificação das atrações também pode ser reaplicada em trabalhos futuros, sua escalabilidade porém ainda não foi aferida e pode se tornar um empecilho para a expansão da base do m.knob para outras cidades.

Por fim, este trabalho apresentou uma forma de se utilizar uma base de conhecimento FrameNet para o desenvolvimento de um sistema de recomendação. Todavia, a integridade dos resultados obtidos deve ser verificada através de métricas e uma avaliação experimental a fim de garantir sua abrangência e também para identificação de casos de uso diferentes.

Casos de uso, nesse sentido, representam a maneira utilizada pelo usuário para alcançar um objetivo, isto é, como a entrada é construída para obter um determinado tipo de recomendação. A definição de quais objetivos podem ser alcançados está diretamente relacionada a abrangência do sistema de recomendação ou o que ele é capaz de recomendar. Esses dois dados, junto com a avaliação das recomendações em si (que representam a capacidade de interpretação de sentenças do sistema), podem contribuir para a elaboração de mudanças no sistema que impactam diretamente em sua qualidade.

Como trabalho futuro, além da inclusão de dados de outras cidades na base de conhecimento do m.knob, o modelo do sistema recomendação do m.knob pode ser transformado em um modelo híbrido. Atualmente o *feedback* do usuário para os itens recomendados já é armazenado, esses dados podem ser utilizados para a construção de perfis de usuário que permitam a realização de uma filtragem baseada em conteúdo. Espera-se que essa filtragem possa melhorar as recomendações, conseqüentemente reduzindo a necessidade do usuário de informar várias sentenças para receber boas recomendações.

Bibliografia

- AGGARWAL, C. C. *Recommender Systems: The Textbook*. 1. ed. [S.l.]: Springer International Publishing, 2016. ISBN 978-3-319-29657-9, 978-3-319-29659-3.
- BRADEŠKO, L.; MLADENIĆ, D. A survey of chatbot systems through a loebner prize competition. In: *Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies*. [S.l.: s.n.], 2012. p. 34–37.
- COSTA, A. et al. Framenet-based modeling of the domains of tourism and sports for the development of a personal travel assistant application. In: *Proceedings of the LREC 2018 International FrameNet Workshop: Multilingual FrameNets and Constructicons*. [S.l.: s.n.], 2018. p. 6–12.
- COSTA, A. D.; TORRENT, T. T. A modelagem computacional do domínio dos esportes na framenet brasil (the computational modeling of the sports domain in framenet brasil) [in portuguese]. In: *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*. [S.l.: s.n.], 2017. p. 201–208.
- DESROSIERS, C.; KARYPIS, G. A comprehensive survey of neighborhood-based recommendation methods. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 107–144.
- D’HARO, L. F. et al. Clara: a multifunctional virtual agent for conference support and touristic information. In: *Natural Language Dialog Systems and Intelligent Assistants*. [S.l.]: Springer, 2015. p. 233–239.
- FELFERNIG, A. et al. Developing constraint-based recommenders. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 187–215.
- FILLMORE, C. J. et al. Frame semantics. *Cognitive linguistics: Basic readings*, p. 373–400, 1982.
- GARRIDO, P. et al. Smart tourist information points by combining agents, semantics and ai techniques. *Computer Science and Information Systems*, n. 00, p. 29–29, 2016.
- JIANG, R. et al. Component pluggable dialogue framework and its application to social robots. In: *Natural Interaction with Robots, Knowbots and Smartphones*. [S.l.]: Springer, 2014. p. 225–237.
- KIM, S.; BANCHS, R. E. R-cube: a dialogue agent for restaurant recommendation and reservation. In: *IEEE. Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*. [S.l.], 2014. p. 1–6.
- KOREN, Y.; BELL, R. Advances in collaborative filtering. In: *Recommender systems handbook*. [S.l.]: Springer, 2015. p. 77–118.
- LOPS, P.; GEMMIS, M. D.; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 73–105.

- MATOS, E. et al. Constructional analysis using constrained spreading activation in a framenet-based structured connectionist model. In: . [s.n.], 2017. Disponível em: (<https://aaai.org/ocs/index.php/SSS/SSS17/paper/view/15257>).
- NICULESCU, A. I. et al. Sara: Singapore’s automated responsive assistant, a multimodal dialogue system for touristic information. In: SPRINGER. *International Conference on Mobile Web and Information Systems*. [S.l.], 2014. p. 153–164.
- PUSTEJOVSKY, J. The generative lexicon. *Computational linguistics*, MIT press, v. 17, n. 4, p. 409–441, 1991.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 1–35.
- RUPPENHOFER, J. et al. *FrameNet II: Extended theory and practice*. [S.l.]: Institut für Deutsche Sprache, Bibliothek, 2016.
- SILVA, L. S. d. et al. Mmrecommender: arquitetura aberta para sistemas de recomendação. Universidade Federal de Juiz de Fora (UFJF), 2017.
- TORRENT, T. et al. Copa 2014 framenet brasil: a frame-based trilingual electronic dictionary for the football world cup. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*. [S.l.: s.n.], 2014. p. 10–14.
- WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, ACM, v. 9, n. 1, p. 36–45, 1966.