

# Projeto e implementação de assistente pessoal para agendamento de compromissos

Mateus Ramos Vendramini, Ricardo Ciriaco Camargo Imagure,  
Marcos Ribeiro Pereira Barretto

**Resumo** – O processo de agendamento de compromissos envolve a condução de uma negociação entre duas ou mais pessoas e pode ser um processo tedioso para muitos, que leva a resultados não ótimos no agendamento devido à falta de comprometimento das pessoas à tarefa de negociação (tanto em ambientes profissionais como pessoais de cada um).

Para resolver esse problema, foi proposta a implementação de um assistente com uma arquitetura dividida em dois grandes módulos: Semantizador, responsável por fazer o processamento de linguagem natural e criar um objeto com uma semântica que é utilizada pelo segundo módulo; Gerenciadores de Diálogo, responsáveis por conduzir o diálogo, processando as entradas dos usuários e gerando saídas.

Essa solução permitiu que metade dos usuários que testaram o assistente conseguissem atingir seus objetivos e que a coleta das informações básicas do compromisso fosse feita com uma média de 3 turnos. Permitiu também uma taxa de 84% de acerto na identificação de entidades relevantes e de 53% de acerto na identificação das intenções relevantes expressas pelo usuário.

**Palavras-chave** – Linguagem Natural, Sistemas Multiagentes, Processamento de Texto.

## 1 Introdução

O processo de agendamento de compromissos envolve a condução de uma negociação entre duas ou mais pessoas e pode ser um processo tedioso para muitos, que leva a resultados não ótimos no agendamento devido à falta de comprometimento das pessoas à tarefa de negociação (tanto em ambientes profissionais como pessoais de cada um). Nesse processo, horários disponíveis podem não ser aproveitados ou horários ruins podem ser agendados sem necessidade.

Parte do desenvolvimento de chatbots ao longo dos últimos anos se dedicou a esse problema, como pode ser visto com a nova IA da Google, o Duplex (LEVIATHAN; MATIAS, 2019). Um dos principais objetivos com esse tipo de ferramenta hoje é: manter uma conversa natural, permitindo que a pessoa fale como se estivesse conversando com outra pessoa para realizar tarefas específicas.

Com isso em mente, a abordagem apresentada neste trabalho tenta alcançar, em algum grau, dois objetivos: uma experiência de uso confortável para o usuário e uma arquitetura que seja ampliável para mais contextos de conversa.

Entende-se que agendar um compromisso é definir: lista de participantes, horário, natureza do compromisso, data, lugar.

Assim, a negociação de um compromisso acaba quando todos os entes envolvidos concordam sobre esses termos. Este trabalho tem por objetivo a implementação de um sistema para agendamento de reuniões que seja capaz de:

- Receber entradas dos usuários em linguagem natural;
- Processar essas mensagens dadas o contexto do diálogo;
- Enviar a resposta respectiva.

Além disso, para o agendamento da reunião, a ferramenta deve permitir uma forma de negociação entre os participantes. Os pontos a serem acordados entre os participantes são: Tipo do compromisso, local do compromisso, horário do compromisso, data do compromisso, lista de participantes.

## 2 Estado da Arte

O estudo de processamento de linguagem natural e, mais especificamente, de sistemas de diálogo está em um campo de estudo ativo e que possui uma série de avanços recentes. Nesta seção, são apresentados conceitos básicos, trabalhos relevantes para a área e algumas das principais ferramentas que podem ser utilizadas para a construção de assistentes.

### 2.1 Entidades, Intenções e Medidas Semânticas

A entidade semântica é um conceito usado de forma bastante genérica e pode assumir diferentes formas: são palavras, expressões da língua, sentenças ou conceitos que possuem algum valor. No contexto deste trabalho, na maior parte do tempo, ela assume a forma de um substantivo e representa algo de valor para o problema específico.

O conceito de intenção, por sua vez, está intimamente relacionado à verbos e representa o que o usuário deseja com aquilo que está falando ou escrevendo.

Com isso, o objetivo do uso de medidas semânticas é conseguir avaliar a proximidade ou interação entre duas entidades semânticas distintas (palavras ou conceitos). Abaixo, há uma tradução de um trecho presente em HARISPE ; et al., sobre o conceito de medidas semânticas.

"Medidas semânticas: ferramentas matemáticas usadas para estimar a força da relação semântica entre unidades linguísticas, conceito ou instancias, por meio de uma descrição matemática obtida da comparação das informações que suportam seus significados."

Segundo [HARISPE ; et al.](#), das diferentes medidas semânticas, duas se destacam, a "similaridade semântica" ("semantic similarity") e a "relação semântica" ("semantic relatedness"). Nesse trabalho, há especial interesse nas medidas de "similaridade semântica", utilizadas na biblioteca NLTK para Python, apresentada na seção 2.2.

## 2.2 Ferramentas Utilizadas

### A *WordNet*

A WordNet ([WORDNET](#), [Acessado em janeiro de 2019](#)) é uma ontologia de propósito geral ([SLIMANI, 2013](#)). No contexto de bancos de dados, um modelo de ontologia é análogo a um modelo relacional, porém, é utilizado em um contexto mais específico de modelagem do conhecimento humano ([GRUBER, 2009](#)). Na WordNet, as palavras são organizadas de forma interligada, por meio de relações que denotam a proximidade existente entre elas. Por exemplo: relações de sinônimos, como "carro" e "automóvel", relações do tipo "geral-específico", como "carro" e "ferrari" ou do tipo "parte-todo", como "volante" e "carro".

### B *CoGroo*

O CoGroo ([COGROO](#), [Acessado em janeiro de 2019](#)) é um corretor gramatical para a língua portuguesa, acoplável ao LibreOffice e é utilizado como classificador morfológico e sintático.

### C *spaCy*

O spaCy ([SPACY](#), [Acessado em janeiro de 2019](#)) é uma biblioteca para processamento de linguagem natural para Python que oferece diferentes funcionalidades, como classificador gramatical, corretor gramatical e reconhecimento de entidades nominadas, disponibilizando modelos pré-treinados em mais de 40 línguas. Neste trabalho, foram exploradas suas funcionalidades de classificador gramatical e reconhecedor de entidades nominadas.

### D *NLTK*

O NLTK ([XUE NIANWEN; LOPER, 2009](#)) é uma biblioteca para Python utilizada para trabalhar com linguagem natural. O NLTK permite trabalhar com diferentes bancos de dados ([CORPORA](#), [Acessado em janeiro de 2019](#)) linguísticos, incluindo o banco da WordNet.

### E *Watson Assistant*

O *Watson Assistant* ([IBM](#), [Acessado em janeiro de 2019](#)) é um serviço de cloud fornecido pela IBM que oferece uma interface para construir assistentes virtuais, permitindo o treino de modelos já testados para detectar informações relevantes numa frase. Essa ferramenta permite que o usuário utilize uma configuração personalizada ao fornecer exemplos de frases de interesse para treinar o modelo disponibilizado. Neste trabalho, por exemplo, a ferramenta é utilizada para detectar intenções na frase dos usuários.

## 2.3 Trabalhos Relacionados

Na área de *chatbots*, existe uma série de trabalhos relacionados ao *Alexa Prize*. Descrito por [RAM ; et al.](#), o *Alexa Prize* foi uma competição proposta pela Amazon em 2016 com o intuito de fomentar progresso na área de inteligência artificial conversacional. O objetivo final da competição é conseguir conduzir uma conversa com tema livre, por pelo menos 20 minutos de forma coerente e engajada.

Para suportar a tarefa, foi fornecido o *Alexa Skills Kit* (ASK) para realizar a tarefa de converter voz para texto e texto para voz. Além disso, os participantes tiveram acesso à base de usuários da Amazon para testar seus *chatbots*. A avaliação do desempenho foi feita a partir da experiência dos usuários, medindo, nas conversas, a coerência, a relevância, o interesse e a capacidade de manter os usuários engajados.

Em linhas gerais, as equipes utilizaram como arquitetura de solução uma divisão em módulos:

- *Alexa ASR*: Responsável por transformar o *input* de voz em texto, converter em um objeto *JSON* e enviar para um *end-point* no *Amazon Web Services*.
- *Natural Language Understanding (NLU)*: Responsável por interpretar a entrada de voz e extrair a informação relevante do enunciado. A partir da informação extraída, uma resposta é gerada.
- *TTS*: Responsável por converter a frase de saída de texto para voz.

Assim, as equipes focaram seus desenvolvimentos nas ferramentas de NLU, cada uma adotando a sua estratégia. O módulo NLU pode ser dividido em módulos, cada um com uma função determinada.

- *Conversation Intent*: Módulo responsável por identificar a intenção do usuário em conversar com o assistente. Foi feito a partir do acoplamento de um detector de intenções procurando por sentenças do tipo "*Alexa, let's chat*".
- *Técnicas de NLU para Diálogos*: Responsável por identificar a semântica e a sintaxe presentes na mensagem do usuário. Esse módulo encapsula toda a complexidade de se entender a linguagem humana. Quanto mais completo for o trabalho desse módulo, mais fácil se torna o tratamento no gerenciador de diálogos.
- *Commonsense Reasoning*: Módulo responsável por simular inferências baseadas em senso comum. Realizado por meio de dados de diálogos.
- *Modelagem de Contexto e Diálogo*: Módulo responsável por lidar com o diálogo. Normalmente implementado por meio de um gerenciador de diálogos principal e múltiplos outros menores, responsáveis por lidar com tarefas e contextos específicos.
- *Gerador de Respostas*: Módulo responsável por gerar a resposta desejada em linguagem natural, a partir de uma mensagem desejada. Quatro estratégias foram utilizadas: Baseado em regras/modelos, baseados em recuperação, baseados em geração e abordagens híbridas.
- *Ranqueamento de Respostas*: Módulo responsável por escolher, por meio de um ranking, qual das respostas possíveis é a que vai melhorar a experiência do usuário.
- *Rastreador de Tópico da Conversa*: Módulo responsável por identificar o tópico da conversa. Como conversas de tópicos diferentes tem conteúdo diferente, para um correto tratamento das entradas dos usuários é necessário conhecer o tópico da conversa.
- *Detector de Mensagens Inapropriadas e Ofensivas*: Módulo responsável por filtrar saídas com conteúdo impróprio. Necessário pois, boa parte dos dados utilizados para treinar os modelos vem de fontes como fóruns da internet ou legendas, que contém conteúdo inapropriado.

Entre os trabalhos participantes, [LARIONOV ; et al.](#) apresentam uma arquitetura em que o gerenciamento de diálogos baseia-se numa máquina de estados finito. Essa escolha permitiu uma representação relativamente simples do modelo de diálogo proposto, contudo, levou a saídas roteirizadas. O sistema foi dividido nos módulos: *Natural Language Understanding*, *Dialog Manager*, e Gerador de respostas, além de módulos auxiliares como a estratégia de ranqueamento, que seleciona entre as saídas possíveis a melhor e o gerenciador de saídas, que se comunica com as APIs externas.

Já o trabalho de [SATO; FIGUEIREDO](#), fora do *Alexa Prize*, descreve um Sistema de Diálogos para agendamento de compromisso. A solução proposta é composta pelos módulos: *Automatic Speech Recognition* - módulo responsável por receber entradas do usuário em voz e converter para texto escrito. , *Semantizador* - módulo responsável por, a partir de uma entrada de texto, extrair informações relevantes para a tarefa de agendar um compromisso, *Gerenciador de Diálogo* - módulo responsável por processar uma entrada do usuário, atualizando o estado do compromisso e gerar uma mensagem aos usuários, transmitindo a mensagem por meio de um Ato Dialogal, *Gerador de Linguagem Natural* - módulo que recebe um ato dialogal e o converte para linguagem natural, *Text-to-speech (TTS)* - converte uma mensagem de texto para áudio. Para o gerenciador de diálogos da etapa de negociação, foi utilizada uma arquitetura *blackboard* e o foco da negociação foi na definição de um horário. O trabalho conseguiu completar a tarefa de coletar as informações iniciais para o compromisso a ser negociado mas não foram exibidos resultados para a parte de negociação.

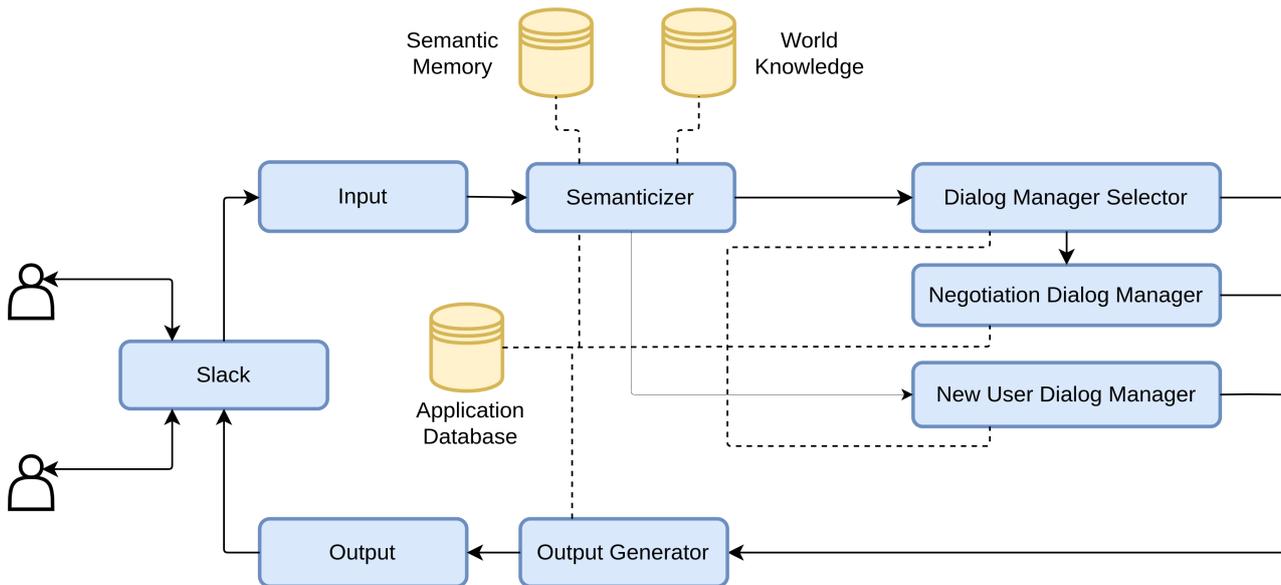
## 3 Arquitetura

### 3.1 Arquitetura Geral

A solução proposta aborda o problema com a proposta de implementação de dois módulos principais, um de *Natural Language Understanding*(NLU) e um de gerenciamento de diálogos. Dentro de cada um desses grandes módulos, é possível implementar diferentes agentes, responsáveis por resolver tarefas específicas. Isso trás um aspecto modular ao sistema, que facilita a integração das diferentes ferramentas utilizadas. Com isso, a arquitetura representada na Figura 1 é proposta, onde os módulos destacados são:

- *Slack*: Módulo de entrada do Slack. Responsável por traduzir comandos da API do Slack para uma estrutura de dados padronizada da aplicação. Pode ser substituído pela integração a outras plataformas, desde que seja mantida a mesma estrutura de dados na saída.
- *Input*: Módulo responsável por gerenciar os diversos módulos de entrada que podem estar presentes na aplicação. Embora na implementação demonstrada apenas uma ponta tenha sido integrada, a arquitetura permite que outras sejam adicionadas.
- *Semantizador*: Módulo responsável por processar a entrada do usuário em linguagem natural e a transformar numa cláusula semântica.
- *Gerenciador de Diálogo Seletor*: Módulo responsável por, a partir da Cláusula Semântica determinar o contexto da conversa, ou seja, a qual compromisso ela se refere. Pode conduzir pequenas conversas com perguntas de desambiguação caso não seja possível determinar o compromisso.

Figura 1 – Arquitetura Geral proposta



Fonte: Autoria Própria.

- Gerenciador de Diálogo de Negociação: Gerenciador de Diálogo principal da aplicação. É responsável por processar as entradas de diversos usuários, atualizando o estado do diálogo a cada interação e gerando respostas coerentes.
- Gerenciador de Novo Usuário: Responsável por conduzir um breve diálogo com o usuário na primeira vez que ele conversar com o assistente. A conversa é bem definida e consiste em perguntar informações como nome completo e solicitar acesso à lista de contatos do usuário.
- Gerador de saída: Módulo responsável por converter as Cláusulas Semânticas de saída dos gerenciadores de diálogo para um texto em linguagem natural.
- Módulo de *Output*: Módulo responsável por realizar a decodificação da saída, enviando a mensagem em linguagem natural para o aplicativo correto.

Para a funcionamento da aplicação, são utilizadas algumas bases de dados. São elas:

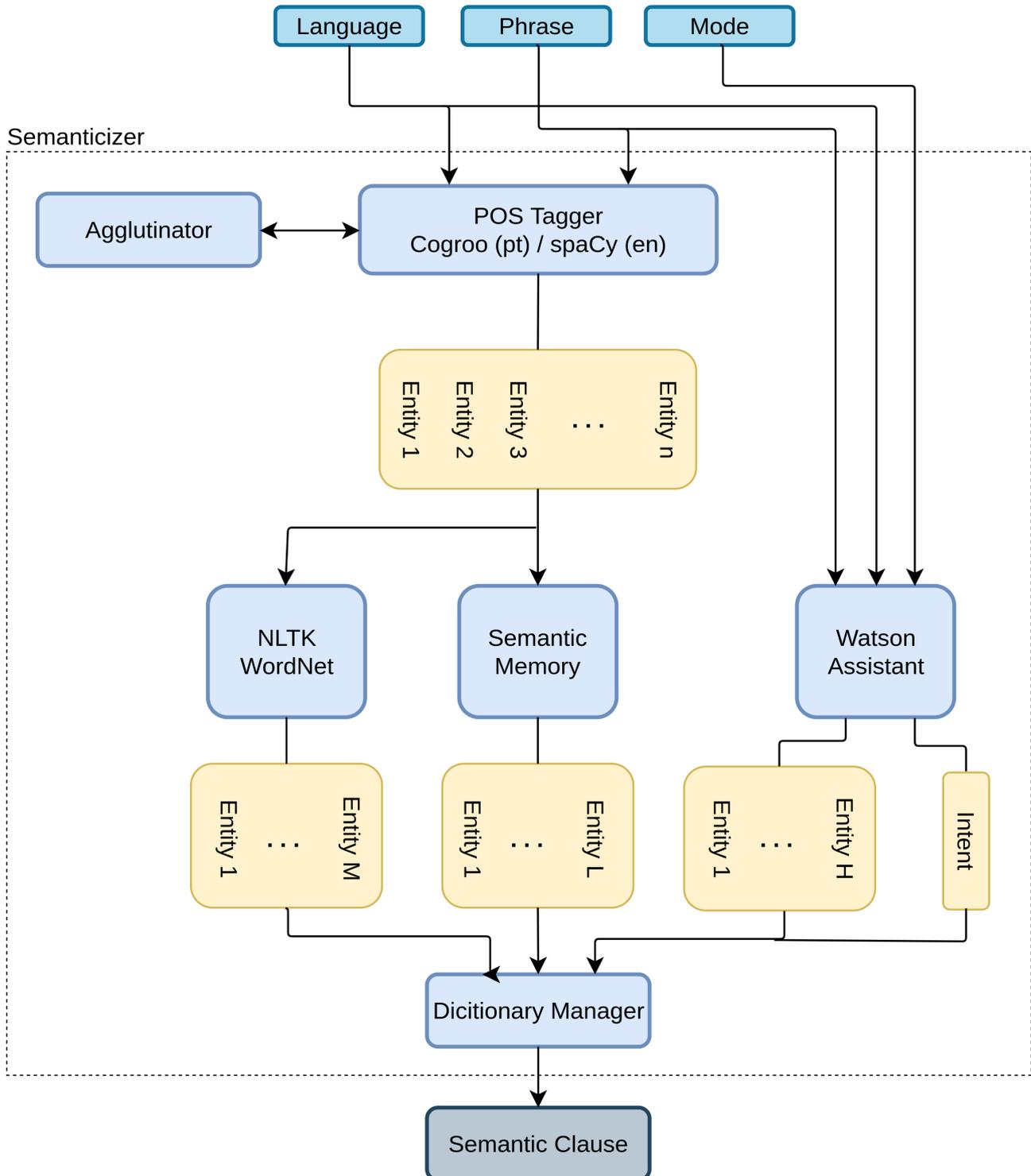
- Memória Semântica: Base de dados personalizável para cada usuário. Contém informações utilizadas para detecção de entidades na entrada do usuário. Foi construída a partir de uma ontologia local e tem, principalmente, a função de determinar as pessoas conhecidas por um usuário.
- *World Knowledge*: Informações genéricas utilizadas para a detecção de entidades baseadas no WordNet. Contém, entre outras coisas, descrições de lugares.
- Banco de dados da aplicação: Base de dados relacional responsável por armazenar dados relativos aos compromissos. Contém, portanto, o estado de cada negociação a cada momento.

### 3.2 Solução Detalhada

#### A Semantizador

Seguindo a abordagem adotada, a arquitetura adotada para o módulo do semantizador é representada como na Figura 2.

Figura 2 – Arquitetura do módulo Semantizador.



Fonte: Autoria Própria.

O Semantizador conta com os seguintes agentes:

- *Watson Assistant Skill*: A IA do *Watson Assistant* é responsável pela classificação de intenções dos usuários, retornando uma medida de confiança para a classificação. E também é responsável pela detecção de datas e horas numa frase.

- *POS Tagger*: O agente *Part of Speech Tagger (POS Tagger)* é o responsável pela classificação morfológica das palavras, muitas vezes associada também à função sintática exercida pela palavra numa sentença. No semantizador, esse agente é responsável por formar uma lista de entidades chamadas de "possivelmente relevantes", que devem ser submetidas aos demais agentes para serem avaliadas. As ferramentas utilizadas para o *POS Tagger* são duas: o corretor gramatical CoGroo, para processar frases em língua portuguesa, e o spaCy, para língua inglesa.
- Aglutinador: O aglutinador é responsável por juntar numa expressão única palavras que podem ter sentido quando juntas. Por exemplo, na frase "Chama o João para o jogo de futebol da quarta-feira", é possível juntar as palavras "jogo", "de" e "futebol" para formarem uma expressão maior que possui sentido, "jogo de futebol". A expressão aglutinada não apenas possui sentido como representa um único tipo de informação. No caso, "jogo de futebol" é uma entidade que representa um tipo de compromisso.
- *Local Ontology*: A memória semântica desenvolvida, no modelo de ontologia, é acessível por meio do agente *Local Ontology*. Através desse agente, é possível encontrar nomes de contatos e lugares conhecidos do usuário.
- *NLTK WordNet*: O corpo de conhecimento é constituído pela base da WordNet, que é acessada por meio da biblioteca *Natural Language ToolKit (NLTK)* do Python. O agente *NLTK WordNet* é o responsável por procurar as entidades "possivelmente relevantes" na *WordNet* e atribuir a elas um atributo "tipo" caso algum significado relevante seja identificado. Os "tipos" procurados na WordNet são lugares, pessoas e tipos de compromissos.
- *Dictionary Manager*: O agente *Dictionary Manager* é responsável por gerenciar os resultados obtidos nos demais agentes, recebendo todas as entidades possivelmente relevantes e definindo a classificação definitiva de seus "tipos".

## B Gerenciador de Diálogos de Negociação

Para esse módulo, adotou-se uma modelagem por meio de uma Máquina de Estados Finitos. O diagrama de estados está exibido na Figura 3. Esse diagrama exhibe o módulo mais alto da máquina de estados. Essa estrutura de Máquina de Estados permite ter controle sobre o modelo do diálogo adotado.

Nessa solução, as informações iniciais do encontro são obtidas a partir de sucessivas interações com o criador do encontro, ator chamado de *meeting owner*. Após coletar o tipo do encontro, a lista de convidados, o local, a data e o horário, inicia-se a fase de negociação.

Na fase negociação, utilizou-se um modelo de diálogo em que o *meeting owner* é o responsável por aceitar ou rejeitar as alterações propostas pelos demais convidados. A negociação termina quando todos os convidados aceitam ou rejeitam o compromisso.

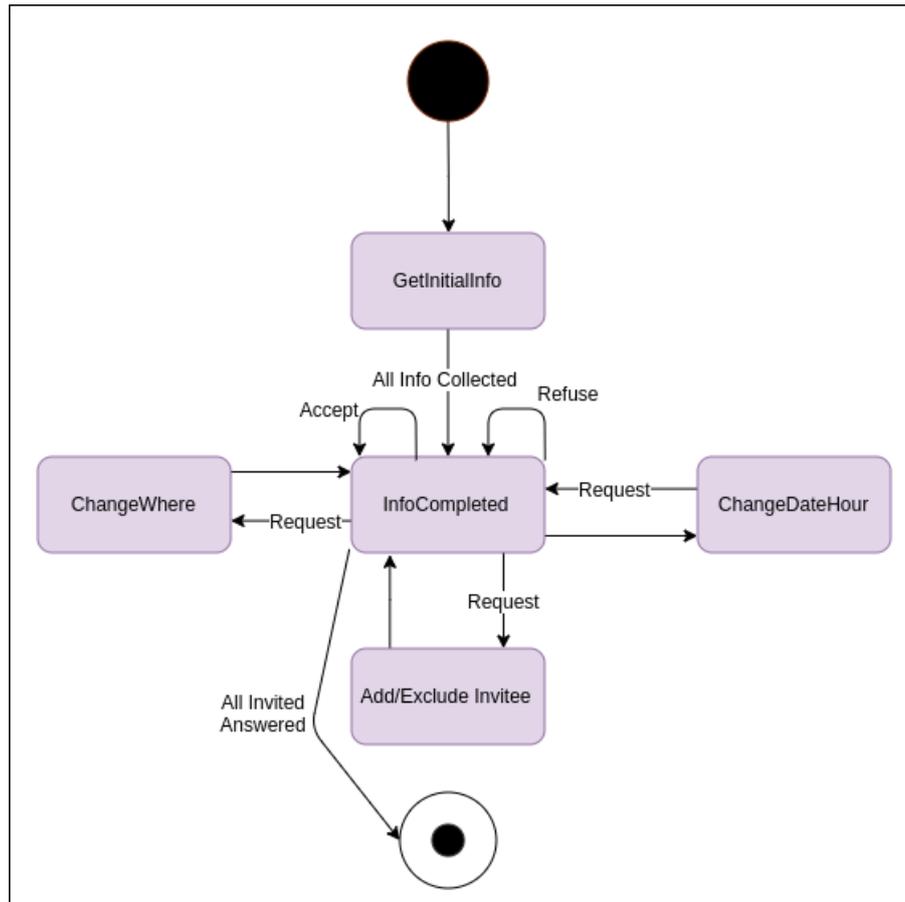
## C Gerenciador de Seleção de Contexto

Esse módulo é o responsável por determinar a qual compromisso um mensagem recebida se refere. Com isso, é possível resolver o caso de uso de alterar dados de um compromisso já marcado.

O fluxograma da figura 4 descreve o comportamento macro desse módulo.

Este é um módulo fundamental para a expansão da arquitetura para novos módulos. Por meio dessa lógica de seleção, determina-se para qual subsistema a mensagem recebida deve ser enviada.

Figura 3 – Máquina de Estados Finitos do Gerenciador de Diálogos. Top level.



Fonte: Autoria Própria.

#### D *Suporte a extensões*

Uma forma de se possibilitar que novos domínios de diálogos possam ser suportados é exibido na Figura 5.

Essa expansão consiste em: 1. Adicionar à lógica do Agente Seletor de Contexto a determinação se a entrada do usuário se refere a esse novo contexto; 2. Criar um novo agente para gerenciar o diálogo desse novo contexto; 3. Eventuais expansões no gerador de saída para suportar novas intenções de saída; 4. Suporte a novas intenções e entidades relevantes no módulo do semantizador.

Com essas alterações, é possível expandir a ferramenta para novos contextos, somente com adições referentes às novas regras de negócio nos módulos anteriores.

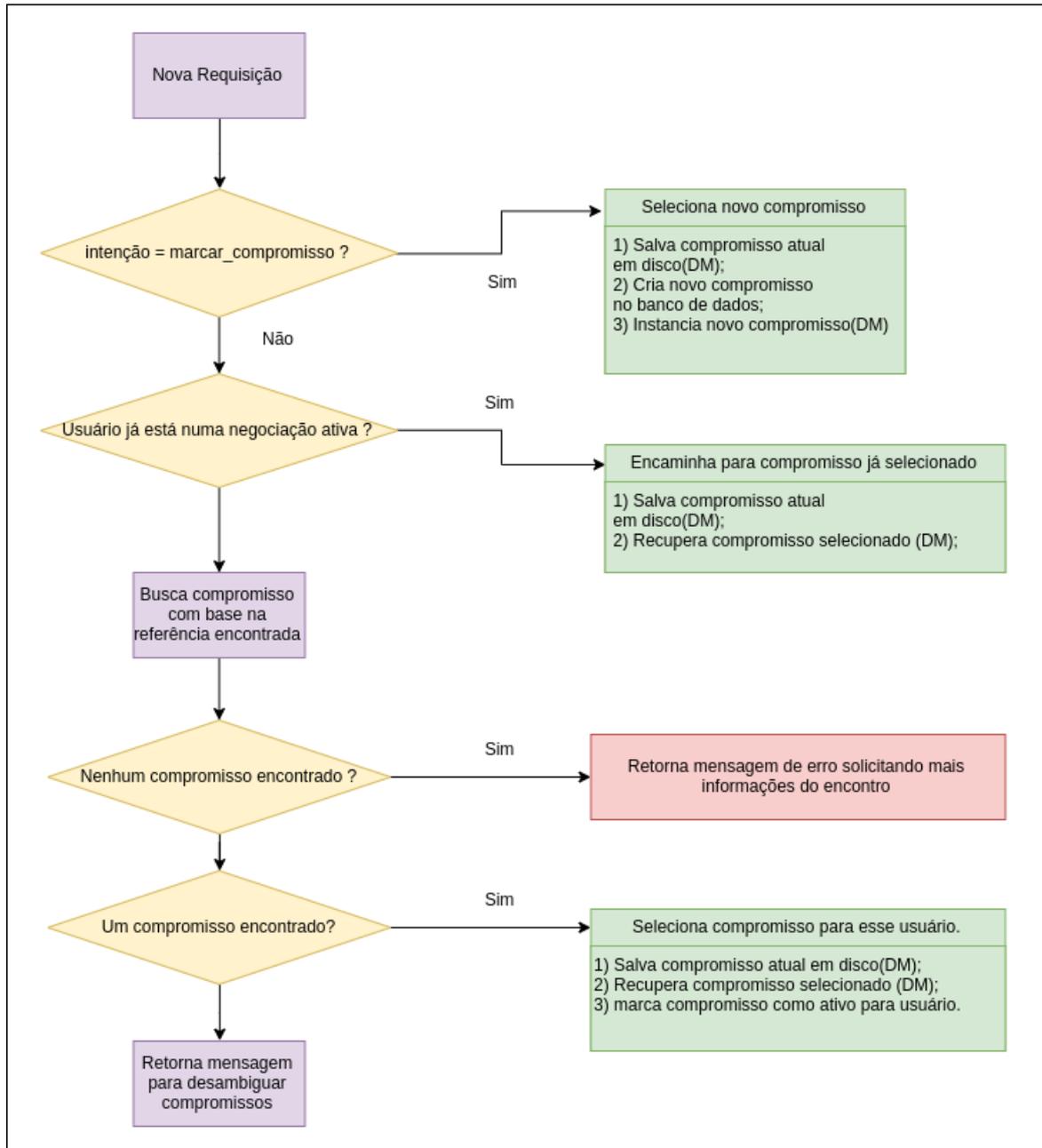
## 4 Resultados

Essa seção é responsável por apresentar os resultados do projeto. Ela é dividida em duas subseções. A primeira demonstra a execução de um dos cenários previstos para o assistente e mostra o resultado das interações de usuários reais com a aplicação. A segunda analisa a aplicação sobre a perspectiva de alguns KPI's selecionados.

### 4.1 Cenário de exemplo

Para esse cenário, os atores envolvidos são:

Figura 4 – Lógica para Agente Seletor de Contexto

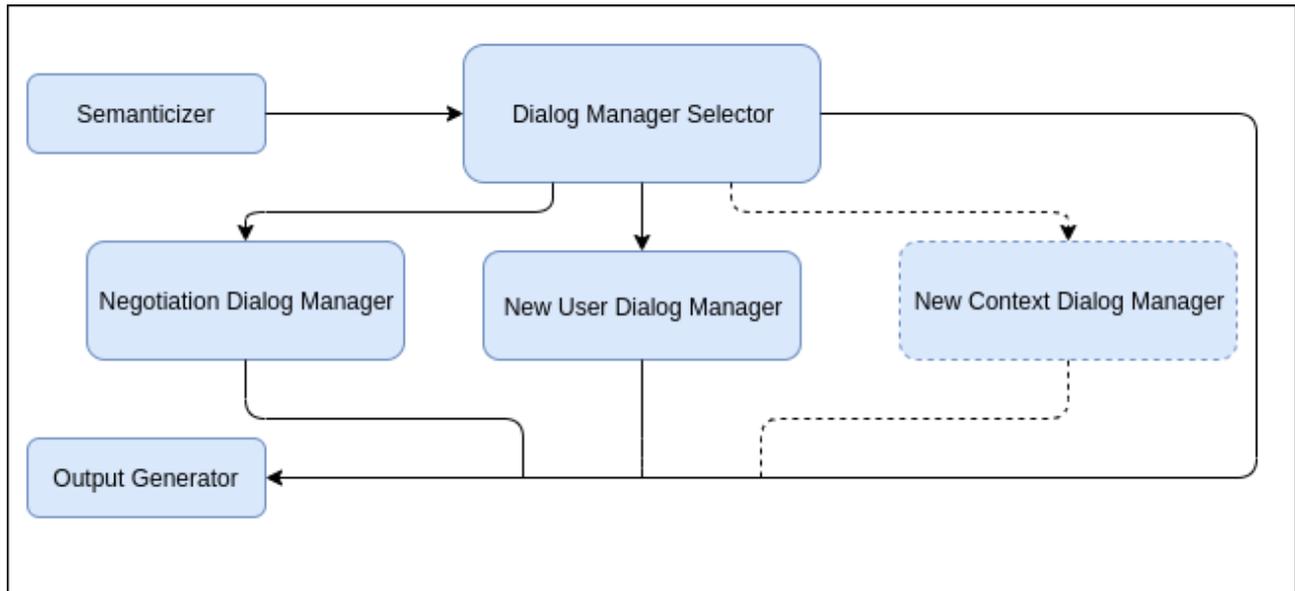


Fonte: Autoria própria.

- Usuário: Usuário genérico que interage com o assistente.
- Assistente: Representa as respostas emitidas pelo sistema de diálogo.
- Convidado: Usuário que foi convidado para um compromisso. Durante os testes com usuários, é chamado de Roberto.
- *Meeting Owner*: Responsável por criar o compromisso que está sendo negociado. Durante os testes com usuários, é chamado de Marcos.

Para padronizar os testes com diversos usuários foi definido um protocolo para que fosse possível comparar a resposta dos diversos usuários sobre o desempenho do assistente.

Figura 5 – Proposta de expansão para novos contextos



Esse protocolo consiste de um roteiro, em que foi atribuído um papel a cada usuário e foram descritos alguns cenários de agendamento de compromisso que os usuários deveriam tentar cumprir.

Com isso, foi possível que os testes fossem realizados com o mínimo de interação possível dos avaliadores com os usuários e que conseguissem cobrir boa parte dos casos de uso previstos para o assistente.

Foi permitido que os usuários interagissem com a aplicação tanto em inglês quanto em português. Isso fez com que mensagens nos dois idiomas por vezes se misturaram numa mesma conversa.

Alguns dos papéis atribuídos para o teste foram:

#### **Pessoa 1: Marcos**

Objetivo: Marcos, você percebeu que, para continuar o seu trabalho é necessário alinhar com Roberto sobre o final da Sprint. Seus horários disponíveis para essa reunião são todos os dias úteis durante a noite. Como vocês dois trabalham no mesmo lugar, o melhor local para a reunião seria no escritório.

#### **Pessoa 2: Roberto**

Marcos provavelmente irá querer marcar uma reunião com você nessa semana. Como seus dias estão corridos, você tem apenas disponível horários a partir das 20h na terça e na quinta-feira. Com relação a lugar não há restrição.

O primeiro ponto a ser avaliado é a coleta das informações iniciais do compromisso. Esse cenário pode ser feito de forma direta, com poucas interações com o sistema:

**Usuário:** marcar uma reunião com Roberto Oliveira as 13 horas na na terça na escola  
**Assistente:** Vou tentar marcar reunião no dia 2019-11-12 às 13:00:00 com Marcos Oliveira e Roberto Oliveira em escola.

Ou de forma detalhada:

**Usuário:** marcar uma reunião  
**Assistente:** Onde deseja marcar, com quem, que dia e que horas?  
**Usuário:** na terça  
**Assistente:** Com quem, onde e que horas deseja marcar?  
**Usuário:** com Roberto Oliveira na escola as 13 horas  
**Assistente:** Vou tentar marcar reunião no dia 2019-11-12 às 13:00:00 com Marcos Oliveira e Roberto Oliveira em escola.

Nos testes com usuários reais, as interações ocorreram de forma gradual. O ator que participou desse processo foi o Marcos.

Houveram situações em que o processo ocorreu de forma fluída:

**Marcos:** schedule new office night meeting  
**Assistente:** Which day, where, with whom and what you want to schedule?  
**Marcos:** hoje a noite, no escritório.  
**Assistente:** O que deseja marcar e com quem?  
**Marcos:** Roberto  
**Assistente:** O que será o compromisso?  
**Marcos:** Final da sprint  
**Assistente:** O que será o compromisso?  
**Marcos:** conversa  
**Assistente:** Vou tentar marcar conversa no dia 2019-10-23 às 18:00:00 com Marcos Oliveira e Roberto Camargo em escritório.

Em outras ocorreu uma série de desentendimentos, em que o usuário precisou alterar significativamente o conteúdo do compromisso e o idioma durante a interação:

**Marcos:** i want to schedule a meeting with Roberto  
**Assistente:** Which day, where and which hour you want to go?  
**Marcos:** on Setis  
**Assistente:** I don't know who Setis is. Please, add him/her to your contacts.. Which hour and place you want to go?  
**Marcos:** on Setis  
**Assistente:** I don't know who Setis is. Please, add him/her to your contacts.. Which hour and place you want to go?  
**Marcos:** thursday on a room  
 Which hour you want to schedule it  
**Marcos:** as 20 horas

## 4.2 Principais Indicadores de Qualidade

Para avaliar a qualidade das interações dos usuários com a ferramenta, foram propostos três índices a serem extraídos dos diálogos: Número de turnos para coleta das informações iniciais; fração de entidades classificadas corretamente; fração de intenções classificadas corretamente.

Para os testes, foram entrevistadas 10 pessoas.

A Tabela 1 exibe os resultados obtidos.

Com estes indicadores é possível ter uma ideia da fluidez do diálogo, já que, em quanto

Tabela 1 – Indicadores de desempenho

Indicador	Valor
Turnos informação Inicial	3
Entidades corretas	84%
Intenções corretas	54 %

menos turnos a coleta de informações for realizada e maior a acurácia da detecção de intenções e entidades, menos interações desnecessárias os usuários têm com a aplicação.

### 4.3 Avaliação dos Usuários

Além dos indicadores, os usuários foram questionados sobre a usabilidade do assistente.

As questões foram:

- 1) O objetivo apresentado foi alcançado? Os compromissos foram marcados com sucesso?
- 2) Você conseguiu entender o que o assistente esperava que você respondesse?
- 3) Você achou que as informações solicitadas foram suficientes, insuficientes ou excessivas para conseguir marcar o compromisso ?
- 4) Dado o número de iterações com o assistente. Você acha que o processo de marcar um compromisso foi facilitado ou dificultado ?

Um resumo dos resultados obtidos é exibido na Tabela 2.

Tabela 2 – Sumário das respostas dos usuários

Questão	Avaliação média
1	metade dos usuários atingiu o objetivo
2	90% dos usuários entendeu o que o aplicativo esperava
3	90% considerou suficiente
4	50% consideraram que o assistente facilitou

## 5 Conclusão

A partir da interação com os usuários, verificou-se que o assistente cumpriu o objetivo de manter uma conversa natural com os usuários, dada a acurácia na identificação das entidades e a avaliação dos usuários, que entenderam o que o assistente desejava.

A arquitetura proposta permite a expansão para outros contextos de diálogos por meio da implementação de novos gerenciadores de diálogos; agentes especializados em identificar entidades e intenções dos novos contextos; suporte a novas intenções de saída e expansão da lógica do seletor de contextos para o novo domínio.

Para trabalhos futuros, destaca-se a possibilidade de desenvolver novos módulos gerenciadores de diálogos e de agentes especializados na identificação de intenções de diferentes contextos. Com essas adições, é possível explorar novos diálogos dentro do problema de agendamento de compromissos e também é possível expandir a ferramenta para novos domínios, como agendamentos de consultas médicas ou negociação sobre compra de produtos em plataformas de vendas.

Percebeu-se que a identificação correta das intenções é crucial para a experiência do usuário e que é necessário um volume maior de interações com pessoas para calibrar os agentes escolhidos. Com a melhora desse ponto, é possível obter um assistente mais amigável, já que o número de situações em que o sistema vai responder de forma inadequada a uma entrada do usuário é menor.

## AGRADECIMENTOS

Ao Professor Marcos Ribeiro Pereira Barretto pelo acompanhamento realizado ao longo do trabalho.

À todas as pessoas que se disponibilizaram para participar dos testes realizados ao longo deste trabalho.

À FDTE pelo seu apoio na disponibilização de recursos físicos para a execução do trabalho e no suporte intelectual que também foi dado.

## REFERÊNCIAS

- COGROO. *Corretor Gramatical Acoplável ao LibreOffice*. Acessado em janeiro de 2019. Disponível em: <<http://cogroo.org/>>. Citado na página 3.
- CORPORA, NLTK. *WordNet, a Lexical Database for English*. Acessado em janeiro de 2019. Disponível em: <<https://www.nltk.org/book/ch02.html>>. Citado na página 3.
- GRUBER, Tom. Ontology. In: \_\_\_\_\_. *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009. p. 1963–1965. ISBN 978-0-387-39940-9. Disponível em: <[https://doi.org/10.1007/978-0-387-39940-9\\_1318](https://doi.org/10.1007/978-0-387-39940-9_1318)>. Citado na página 3.
- HARISPE, Sébastien ; et al. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers LLC, v. 8, n. 1, may 2015, 6–20. Disponível em: <<https://doi.org/10.2200%2Fs00639ed1v01y201504hlt027>>. Citado 2 vezes nas páginas 2 e 3.
- IBM. *Watson Assistant*. Acessado em janeiro de 2019. Disponível em: <<https://www.ibm.com/cloud/watson-assistant/>>. Citado na página 3.
- LARIONOV, George ; et al. Tartan: A retrieval-based socialbot powered by a dynamic finite-state machine architecture. *arXiv e-prints*, [S.l.: s.n.], Dec 2018, arXiv:1812.01260. Citado na página 5.
- LEVIATHAN, Yaniv; MATIAS, Yossi. *Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone*. 2019. Disponível em: <<https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>>. Citado na página 2.
- RAM, Ashwin ; et al. The science behind the alexa prize. *Alexa.Prize.Proceedings*, [s.n.], 2018. Disponível em: <[https://developer.amazon.com/alexaprize/proceedingsaccessed\(2018\)-01-01](https://developer.amazon.com/alexaprize/proceedingsaccessed(2018)-01-01)>. Citado na página 3.
- SATO, Erich; FIGUEIREDO, Nicolas. Sistema de diálogo para agendamento automático de compromissos. [S.l.: s.n.], 2017. Citado na página 5.
- SLIMANI, Thabet. Description and evaluation of semantic similarity measures approaches. *International Journal of Computer Applications*, Foundation of Computer Science, v. 80, n. 10, oct 2013, 25–33. Disponível em: <<https://doi.org/10.5120%2F13897-1851>>. Citado na página 3.
- SPACY. *Industrial-Strength Natural Language Processing*. Acessado em janeiro de 2019. Disponível em: <<https://spacy.io/>>. Citado na página 3.
- WORDNET. *WordNet, a Lexical Database for English*. Acessado em janeiro de 2019. Disponível em: <<https://wordnet.princeton.edu/>>. Citado na página 3.
- XUE NIANWEN, Steven Bird Evan Klein; LOPER, Edward. Natural language processing with python. [S.l.]: O'Reilly Media Inc, n. 3, 2009. Citado na página 3.

**Title** - Project and implementation of a personal assistant for meeting scheduling

**Abstract** – The process of scheduling a meeting relies on a negotiation between two or more actors and might be a tedious task, therefore resulting in unoptimized results given the lack of commitment of the actors.

To solve this problem, it was proposed an assistante based on an architecture with two main modules: Semanticizer, responsible for the natural language processing and the creation of an object that represents the semantic of the user utterance; Dialog Managers, responsible for dialog's conduction, processing the Semanticizer output and generating outputs.

This solution allowed half of the testers to accomplish their goals while using the assistant, with an average of three turns to collect all relevant data to the meeting. It also achieved 84% of accuracy to identify relevant entities and 53% of accuracy to identify relevant intents expressed by the users.

**Keywords** – Natural Language, Multi-agent Systems, Text Processing.

**Mateus Ramos Vendramini** Aluno do último semestre da Engenharia Mecatrônica.

**Ricardo Ciriaco Camargo Imagure** Aluno do último semestre da Engenharia Mecatrônica.