

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

GABRIEL OKAMOTO CARLOS

Desenvolvimento de um chatbot sobre a Amazônia Azul

São Paulo

2021

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

GABRIEL OKAMOTO CARLOS

Desenvolvimento de um chatbot sobre a Amazônia Azul

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Mecatrônico e de Sistemas Mecânicos.

São Paulo

2021

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

GABRIEL OKAMOTO CARLOS

Desenvolvimento de um chatbot sobre a Amazônia Azul

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Mecatrônico e de Sistemas Mecânicos.

Orientador:

Fabio Gagliardi Cozman

São Paulo

2021

AGRADECIMENTOS

Ao professor Fabio Cozman, pela orientação, apoio e condução desse trabalho.

Aos integrantes do Blue Amazon Brain (BLAB), pelo apoio e sugestões.

À Universidade de São Paulo, pelo ambiente de qualidade e oportunidade de realizar o curso de Engenharia Mecatrônica.

Aos docentes, direção e administração do curso, pela qualidade do ensino fornecido.

À minha mãe, meu pai e minha madrasta, pelo amor incondicional, incentivo e apoio.

Aos meus amigos, que me trouxeram até aqui.

À minha namorada, pela compreensão, incentivo e apoio.

RESUMO

A Amazônia Azul, nome conferido à região marítima brasileira, possui um enorme valor econômico, científico e social para o Brasil, porém não recebe o devido reconhecimento pelo público geral. Com o objetivo de ampliar a educação, o conhecimento e o interesse nesse tema, desenvolveu-se um chatbot capaz de responder perguntas sobre os diversos tópicos contidos, como preservação e recursos vivos, energéticos e minerais. Obteve-se uma base de perguntas filtrando-se dados de perguntas reais feitas na internet e elaborou-se a base de um framework para a geração de chatbots de perguntas e respostas. O chatbot produzido foi testado e avaliado por voluntários. Os resultados indicaram uma boa recepção e a viabilidade do chatbot como ferramenta para a educação nesse tema.

Palavras-chave: chatbot; Amazônia Azul; educação; Topic Modelling; Dialog State Tracking.

ABSTRACT

The Blue Amazon, the name given to the Brazilian maritime region, has enormous economic, scientific and social value for Brazil, but it does not receive due recognition by the general public. Aiming to expand education, knowledge and interest in this matter, a chatbot was developed capable of answering questions about the various topics contained in it, such as preservation and living, energy and mineral resources. A database of questions was obtained by filtering data from real questions asked on the internet and the basis for a framework for the generation of Question Answering chatbots was devised. The chatbot produced was tested and evaluated by volunteers. The results indicated a good reception and the viability of the chatbot as a tool for education in this subject.

Keywords: chatbot; Blue Amazon; education; Topic Modelling; Dialog State Tracking.

SUMÁRIO

INTRODUÇÃO	10
1 Introdução	11
1.1 Motivação	12
1.2 Estrutura do trabalho	13
OBJETIVOS	14
2 Objetivos.....	15
CONCEITOS-CHAVE	16
3 Conceitos-chave.....	17
3.1 Tempo de treinamento	17
3.2 Embedding.....	17
3.3 Topic Modelling	18
3.1 Valor-p.....	18
3.2 Clustering.....	18
3.3 Term Frequency-Inverse Document Frequency	18
3.1 Framework.....	19
3.2 Conceitos de programação.....	19
ESTADO DA ARTE	20
4 Estado da arte.....	21
4.1 Conceitos principais	21
4.1.1 Chatbots orientados a tarefas versus orientados a diálogo	21
4.1.2 Intenção	21
4.1.3 Entidades	22
4.1.4 Elocuções	22

4.1.5	Diálogo linear ou não-linear.....	23
4.1.6	Contexto	23
4.2	Estudos relacionados	24
4.3	Revisão sistemática de literatura	25
4.4	Comparação de frameworks	25
4.5	Watson Assistant	26
MÉTODOS DE AVALIAÇÃO DE RESULTADOS		28
5	Métodos de avaliação de resultados	29
METODOLOGIA		31
6	Metodologia.....	32
6.1	Requisitos	32
6.1.1	Requisitos não-funcionais	32
6.1.2	Requisitos funcionais	32
6.2	Desenvolvimento do chatbot	34
6.2.1	Obtenção de perguntas	36
6.2.2	Classificação das perguntas.....	38
6.2.3	Arquitetura dos nós de diálogo.....	42
6.2.4	Implementação em código	46
6.2.4.1	Geração de intenções	47
6.2.4.2	Geração de entidades	48
6.2.4.3	Geração de nós de diálogo.....	48
6.2.4.4	Organização dos nós de diálogo	50
6.3	Distribuição do chatbot.....	51
RESULTADOS		53
7	Resultados.....	54

CONCLUSÃO	58
8 Conclusão	59
8.1 Considerações para trabalhos futuros	59

INTRODUÇÃO

1 Introdução

Amazônia Azul (AAz) é o nome conferido à região marítima definida pela Zona Econômica Exclusiva (ZEE) brasileira junto da Extensão da Plataforma Continental, totalizando uma extensão estimada em 4,5 milhões de quilômetros quadrados, mais que a metade da área continental do Brasil (Castro et al., 2017). O nome faz referência à Amazônia devido à riqueza de recursos vivos, minerais e energéticos e de biodiversidade encontrados na região. A AAz, no entanto, não recebe o mesmo reconhecimento que a sua semelhante continental, de modo que seu potencial econômico e científico não é completamente aproveitado.

O problema proposto é o de desenvolver um agente conversacional sobre a AAz: um chatbot com capacidade de responder perguntas complexas sobre a região oceânica da costa brasileira. Os termos chatbot, agente conversacional e robô serão utilizadas de maneira intercambiável ao longo desse trabalho e representam sistemas de diálogo que processam diálogo natural e respondem automaticamente com linguagem humana (Deep AI, 2019).

Para entender a situação do conhecimento acerca da AAz e quais informações possuem maior demanda, realizou-se uma análise de perguntas feitas por pessoas na internet, filtrando dados disponibilizados pelo Bing, um motor de busca popular da Microsoft. O processo de filtragem de dados não-rotulados é uma tarefa notoriamente complexa, que foi abordada nesse trabalho através de algoritmos estatísticos e de aprendizado de máquina realizados na língua de programação Python. Uma vez obtidas as perguntas, pesquisou-se as respostas em fontes públicas e confiáveis de informação. Com o banco de dados montado, elaborou-se um algoritmo para programar o chatbot de maneira eficiente com essas informações. Esse algoritmo foi programado também em Python.

O robô foi disponibilizado em um site próprio, divulgado para uma ampla audiência e avaliado por usuários através de um formulário. O formulário incluía perguntas para medir fatores de qualidade de chatbots encontrados na literatura, como a satisfação geral do usuário, assim como indicadores do sucesso do robô como uma ferramenta de educação, por exemplo, o nível percebido de confiabilidade das respostas e o quanto o usuário considera ter aprendido com a experiência.

Como base para o desenvolvimento, utilizou-se o Watson Assistant, ferramenta da IBM para desenvolvimento de chatbots que possui entendimento de linguagem natural (*Natural Language Understanding* – NLU) baseada em inteligência artificial. O Watson permite criar chatbots com pouco a nenhum conhecimento prévio de programação, através de uma

interface interativa em uma página web, e com diversas funcionalidades para providenciar a melhor experiência para o usuário.

1.1 Motivação

A AAz possui um enorme potencial econômico e científico. Ela possibilita atividades como pesca, turismo, transporte marítimo, extração de petróleo e gás e geração de energia renovável. Segundo a Marinha do Brasil, mais de 95% do comércio exterior brasileiro é escoado por rotas marítimas e cerca de 85% do petróleo, 75% do gás natural e 45% do pescado produzidos no país saem da AAz (Pereira, 2019).

A importância da AAz, porém, não é compreendida completamente pela população brasileira. Ela é reconhecida “principalmente por ser fonte de alimentos e de lazer” (Castro et al., 2017), uma visão reduzida de seu verdadeiro potencial.

Nesse contexto, existe uma lacuna de educação que pode ser preenchida por um agente conversacional. Ele pode servir como uma maneira central para divulgar informações e conhecimento e instigar curiosidade e interesse acerca do tema.

Conversas humanas possuem diversas camadas de complexidade (sintaxe, ortografia, semântica etc.) que, juntas, impõem um enorme desafio à interpretação pelos computadores. Ferramentas como o Watson já existem com o objetivo de facilitar essa tarefa, por exemplo a detecção de intenção. “Qual é a extensão territorial da Amazônia Azul?”, “Qual é a área ocupada pela Amazônia Azul” e “Quantos quilômetros quadrados ocupa a Amazônia Azul?” são todas perguntas que se referem a um mesmo fato – logo, possuem a mesma intenção – redigidas de maneiras alternativas. O Watson é bem equipado para entender isso. Porém, ainda cabe ao desenvolvedor programar respostas a situações complexas. Imagine que um usuário pergunte “Quantas espécies há na Amazônia Azul?” e, em seguida, “Quantas estão em extinção?”. Note que a segunda pergunta sozinha não é completa, pois faz referência à anterior. Logo, é necessário guardar o contexto da conversa, que, nesse caso, pode ser resumido ao tópico da última pergunta. Assim, seria possível identificar a intenção da segunda pergunta e redirecionar para a resposta adequada. Esse obstáculo técnico foi descrito por Williams et al. (2013) como *dialog state tracking* (acompanhamento do estado do diálogo) e múltiplas técnicas foram desenvolvidas para contorná-lo. Esse desafio, entretanto, vai além do escopo do presente trabalho e foi apenas parcialmente investigado. Adotou-se a estratégia de separar as perguntas por tópico principal (por exemplo, tartarugas, baleias ou petróleo) e memorizar o tópico da última

pergunta feita, o que permite o usuário fazer perguntas em sequência sobre um mesmo tema sem precisar repeti-lo cada vez (assim como no exemplo anterior).

1.2 Estrutura do trabalho

As próximas seções desse trabalho estão estruturadas da seguinte forma:

2. Objetivos: o que se deseja obter ao final do trabalho.
3. Conceitos-chave: explicação de conceitos-chave para o trabalho.
4. Estado da arte: análise de trabalhos recentes relacionados ao tema.
5. Métodos de avaliação de resultados: análise de quais métodos são mais utilizados nesse campo para determinar a qualidade do resultado.
6. Metodologia: explicação da metodologia adotada, incluindo a definição dos requisitos do projeto.
7. Resultados: apresentação dos resultados coletados (dados do chatbot e avaliações de usuários) e análise frente aos requisitos e objetivos.
8. Conclusão: contribuições do trabalho e considerações para trabalhos futuros.

OBJETIVOS

2 Objetivos

O objetivo desse trabalho é desenvolver um chatbot capaz de responder perguntas informativas acerca da Amazônia Azul, utilizando informações extraídas de textos, artigos e livros da internet. Ele deve ser capaz de cobrir aspectos básicos do tema, como recursos vivos, minerais e energéticos, de modo a informar e instigar a curiosidade. Tendo em vista a educação e o fomento ao interesse no tema como objetivos primários, é essencial medir o nível de satisfação com a experiência de quem testar o robô, assim como outros fatores, dentre eles: nível de aprendizado, nível de engajamento, nível de confiabilidade das informações etc. Ao fim, deseja-se que cada uma das métricas possua pontuação acima da média, assim como a média geral entre elas seja positiva também, indicando o sucesso do trabalho. O público-alvo é composto de brasileiros com ensino fundamental completo, algum interesse por ciência, que não estudem ativamente um tema relacionado (por exemplo, graduandos de cursos de biologia) e que possuam aptidão com tecnologia.

Como objetivo secundário, o chatbot deve ser capaz de responder não só perguntas simples, mas também complexas, como: perguntas que dependem de contexto; perguntas que envolvem diferentes fontes de informação; perguntas ambíguas. Alguns exemplos estão no Quadro 1.

Quadro 1 – Exemplos de perguntas simples e complexas para um chatbot

Perguntas simples	Perguntas complexas
Quantas espécies de vida marítima existem na Amazônia Azul?	Quantas espécies de vida marítima existem na Amazônia Azul? [seguido de...] Quais delas estão em extinção?
Quantas reservas de petróleo há na Amazônia Azul?	Por que a Amazônia Azul é importante para o Brasil?
Qual é a extensão territorial da Amazônia Azul?	Qual é a participação da extração de petróleo no PIB brasileiro?

Fonte: Autor

CONCEITOS-CHAVE

3 Conceitos-chave

Alguns dos conceitos utilizados ao longo desse trabalho são brevemente explicados nesse capítulo, divididos nas categorias:

- Aprendizado de máquina
 - Tempo de treinamento: usado para comparar ferramentas de criação de chatbots na Seção 4.4
 - Embedding: usado na filtragem do banco de dados, na Seção 6.2.1
 - Topic Modelling: usado na filtragem do banco de dados, na Seção 6.2.1
- Estatística
 - Valor-p: usado em testes de hipóteses na literatura e na Seção 0
 - Clustering: usado na filtragem do banco de dados, na Seção 6.2.1
 - Term Frequency-Inverse Document Frequency (TF-IDF): usado na filtragem do banco de dados, na Seção 6.2.1
- Programação
 - Framework: usado para descrever ferramentas de código em diferentes contextos
 - Conceitos de programação: usados para descrever o código produzido na Seção 6.2

3.1 Tempo de treinamento

O tempo de treinamento de um algoritmo de aprendizado de máquina é definido como o necessário para ele processar um trecho definido de um conjunto de dados (por exemplo, os primeiros 10%) e criar um modelo capaz de operar sobre o conjunto completo. Dependendo do tamanho do banco de dados e da complexidade do algoritmo, esse tempo pode ser da escala de segundos, minutos, horas ou até mesmo dias, tornando-o uma métrica importante na comparação de algoritmos.

3.2 Embedding

Um *embedding* é o mapeamento de uma variável discreta para um vetor de números contínuos (Koehrsen, 2018). No caso de texto, conjuntos de palavras de comprimentos variáveis são convertidos em sequências de números de comprimento fixo, a partir dos quais é possível realizar agrupamentos matematicamente.

Esse conceito é amplamente utilizado por algoritmos de aprendizado de máquina para operar com textos, pois permite que eles sejam tratados como números de uma forma que busca preservar relações semânticas.

3.3 Topic Modelling

O *Topic Modelling* é uma técnica estatística amplamente utilizada no campo de mineração de texto. O objetivo é, dado um conjunto de documentos, automaticamente identificar tópicos e a distribuição deles entre os documentos, sendo um tópico definido como um conjunto de palavras recorrentes (Blei et al., 2010; Bansal, 2016).

No presente trabalho, essa técnica é utilizada na filtragem do banco de dados de perguntas feitas no Bing.

3.1 Valor-p

O valor-p é uma medida da estatística clássica utilizada em testes de hipótese, ou seja, para determinar se uma hipótese é válida com base em dados estatísticos. Suponha um teste com as hipóteses nula (H_0) e alternativa (H_A), sendo, por convenção, H_A aquela que se deseja provar e H_0 , seu complemento. O valor-p é definido como a probabilidade de se observar um conjunto de dados igualmente ou mais extremos que o da amostra analisada, assumindo como verdadeira H_0 . Um dado “extremo” é aquele que tende a rejeitar H_0 e confirmar H_A .

O valor-p é usado, então, para aferir a significância estatística de um resultado. Quanto menor o valor, mais provável é a hipótese desejada. O valor mais comumente encontrado no meio científico e acadêmico é de 0,05, seguido do valor de 0,01.

3.2 Clustering

Clustering é a tarefa de agrupar objetos em *clusters*, baseado na similaridade em algum aspecto. Objetos pertencentes a um *cluster* são mais similares entre si que quando comparados a objetos de outros *clusters*. Algoritmos de *clustering* são utilizados nesse trabalho como parte da tarefa de *Topic Modelling*.

3.3 Term Frequency-Inverse Document Frequency

O *Term Frequency-Inverse Document Frequency* (TF-IDF) é uma medida da frequência de palavras em documentos (Rajaraman & Ullman, 2011). A primeira parte (TF) é o cálculo da frequência relativa da palavra no documento (1 para a palavra mais frequente e uma fração

disso para as demais). A segunda parte (IDF) é o cálculo do inverso da frequência da palavra entre os documentos do conjunto (quão mais frequente a palavra, menor esse valor). A combinação dessas duas medidas gera uma pontuação indicativa do quanto a palavra representa o documento.

No presente trabalho, essa medida é utilizada para determinar palavras representativas de um *cluster* de frases, permitindo rapidamente identificar seu tópico.

3.1 Framework

No contexto de programação, *frameworks* são uma abstração em que software de funcionalidade genérica podem ser estendidos pelo usuário, assim criando software específicos de aplicação (Riehle, 2000). *Frameworks* podem incluir programas, ferramentas, APIs, bibliotecas de código e outros componentes que habilitam o desenvolvimento de um projeto ou sistema.

3.2 Conceitos de programação

Em algumas etapas desse trabalho, utiliza-se a linguagem Python de programação. Nesse contexto, alguns termos possuem significado especial:

- Lista (list): conjunto mutável e ordenado de elementos
- Dicionário (dict): conjunto mutável e não-ordenado de pares chave-valor
- Set: conjunto imutável e não-ordenado de elementos únicos
- Booleano (bool): valor que pode ser verdadeiro ou falso
- String: sequência de caracteres
- Application Programming Interface (API): um conjunto de regras que definem como computadores ou aplicações se comunicam entre si (IBM, 2020)
- Função/método: trecho de código que recebe uma instrução e realiza um procedimento, podendo possuir entrada e/ou saída de dados
- Classe: na programação orientada a objetos, classes são *templates* (modelo/gabareto) para a criação de objetos. Elas definem valores iniciais e comportamentos através de métodos

Observação: a maior parte dos conceitos não é específica dessa língua, porém suas definições podem variar em outros contextos.

ESTADO DA ARTE

4 Estado da arte

Em 1966, o primeiro chatbot, chamado Eliza, foi desenvolvido pela equipe de Joseph Weizenbaum, no *Massachusetts Institute of Technology*. Usuários podiam mandar curtas mensagens para o computador, que então as interpretava através de um reconhecimento de padrões rudimentar e respondia com uma de suas respostas pré-programadas, misturando-a com informações do texto de entrada. Isso fornecia aos usuários um sentimento de ser ouvidos e entendidos (Weizenbaum, 1966, p. 6).

Desde então, diversas tecnologias foram desenvolvidas para auxiliar no desafio de interpretar a fala humana, capacidade conhecida como entendimento de linguagem natural (*Natural Language Understanding* – NLU), um subtópico do processamento de linguagem natural (*Natural Language Processing* – NLP). NLU e NLP podem ser considerados sinônimos no contexto desse trabalho e dos trabalhos mencionados.

Os recentes avanços da inteligência artificial e do aprendizado de máquinas, assim como de poder de processamento dos computadores, foram fatores cruciais para o desenvolvimento de chatbots cada vez mais capazes de entender a fala humana e responder à altura.

4.1 Conceitos principais

Após anos de pesquisa e prática no campo de chatbots, alguns conceitos foram estabelecidos e são comuns a diferentes implementações. Eles são explicados a seguir.

4.1.1 Chatbots orientados a tarefas versus orientados a diálogo

Chatbots orientados a tarefas são aqueles que ajudam usuários a cumprir determinadas tarefas, como comprar passagens de avião, reservar quartos de hotel ou mesas de restaurante, realizar compras online etc. Já os orientados a diálogo são aqueles cujo objetivo é conversar com o usuário, seja para fins educativos ou de entretenimento, por exemplo.

4.1.2 Intenção

A intenção é o objetivo do usuário com uma mensagem. Por trás de cada texto enviado ao chatbot, o usuário espera algo em retorno, seja uma resposta ou uma ação. Alguns exemplos de possíveis pares pergunta-intenção estão no Quadro 2. Em todos os exemplos, a ação é “responder”, pois o chatbot do presente trabalho é orientado a diálogo, mas isso não se aplica sempre.

Quadro 2 – Exemplos de perguntas e as respectivas intenções

Pergunta	Intenção (ação: descrição)
Quantas reservas de petróleo há na Amazônia Azul?	Responder: petróleo/quantidade de reservas
O que é Amazônia Azul?	Responder: amazônia azul/definição
Quanta pesca é feita no oceano anualmente?	Responder: pesca/quantidade

Fonte: Autor

4.1.3 Entidades

Entidades são dados adicionais que podem ser extraídos do texto de entrada, além da intenção. Usos mais comuns são datas, valores monetários, nomes próprios etc. É importante tratar o conteúdo deles separadamente da mensagem, pois eles não devem influenciar na detecção da intenção. Por exemplo, os comandos “Comprar uma passagem de avião no dia 17 de maio” e “Comprar uma passagem de avião no dia 29 de julho” devem acionar a detecção da mesma intenção, apesar de possuírem datas diferentes, e o valor dessa data deve ser usado pelo robô na execução da tarefa.

4.1.4 Elocuções

Elocuções (do inglês: *utterances*) são diferentes maneiras de escrever uma mesma mensagem. São variações de forma, porém não de conteúdo. Alguns exemplos estão no Quadro 3.

Quadro 3 – Exemplos de elocuções e respectivas intenções

Elocuções	Intenção (curta descrição)
Quantas espécies há na Amazônia Azul? Qual é a quantidade de espécies na Amazônia Azul?	fauna/quantidade de espécies
Qual é a área de Amazônia Azul? Qual é a quilometragem da Amazônia Azul? Quantos metros quadrados a Amazônia Azul ocupa?	amazônia azul/detalhar tamanho

Fonte: Autor

Um detalhe importante na língua portuguesa é a variedade de forma. Diferenças como abreviações (“você” versus “vc”, “por que” versus “pq”), acentuação (“é” versus “eh”) e características regionais (emprego de “tu” ou “você”) são fatores que alteram o entendimento de chatbots e precisam ser levados em conta. No caso do Watson Assistant, mesmo quando configurado para utilizar a língua portuguesa, esses fatores prejudicam severamente o reconhecimento de intenções, sendo necessário cuidado especial.

4.1.5 Diálogo linear ou não-linear

Em um diálogo com um chatbot, é possível que uma interação exija outras subseqüentes para completar uma tarefa. Por exemplo, caso um usuário peça a compra de uma entrada de cinema, é provável que o chatbot necessite informações adicionais, como: local, data, filme etc. Nem sempre o usuário providencia todas elas em uma única declaração. Esse é o caso de um diálogo linear, no qual o robô realiza uma série de perguntas ou confirmações até possuir todos os detalhes que precisa para executar uma ação.

O diálogo linear é útil em situações nas quais existe um conjunto definido de informações necessárias. O caso geral e que mais se aproxima de uma conversa natural é o não-linear, especialmente no caso de um robô destinado a responder perguntas, não realizar tarefas.

4.1.6 Contexto

O contexto é uma parte essencial da comunicação humana. Muitas vezes, assume-se que o receptor da mensagem possui memória do que foi dito anteriormente, o que é normalmente válido para humanos, porém um obstáculo para robôs. Não basta saber o que foi dito, como também é necessário saber interpretá-lo. Suponha a seguinte troca:

- Usuário: Quantas espécies estão em extinção?
- Robô: Um total de [...] espécies estão em extinção na Amazônia Azul.
- Usuário: Quantas delas são peixes?

A segunda pergunta, por si só, não possui informações suficientes para gerar uma resposta. O robô precisa lembrar que o usuário estava perguntando sobre espécies em extinção anteriormente para responder corretamente. Logo, para o funcionamento de um bom chatbot, é necessário guardar a informação do contexto de alguma maneira.

4.2 Estudos relacionados

Diversos estudos foram realizados desde a invenção da internet para avaliar o impacto da tecnologia no ensino à distância. Yin et al. (2021) realizaram um estudo com 99 alunos de primeiro ano de faculdade para comparar o aprendizado tradicional (em uma sala de aula, com um professor) versus através de um chatbot (em um aplicativo, com tratamento individual). Os alunos foram divididos em um grupo de controle (tradicional) e um grupo experimental (chatbot). Em um teste sobre o que aprenderam, verificou-se que a performance dos alunos foi similar entre os grupos, porém o grupo experimental apresentou um desvio padrão significativamente menor, ou seja, alunos nesse grupo tiveram um aprendizado mais uniforme, com menos pontos fora da curva. Também foi observado entre os alunos do grupo experimental motivação intrínseca mais forte, menos stress sofrido e uma melhor percepção da contribuição do ambiente de estudo para o aprendizado. Ambos resultados mencionados possuem valores-p abaixo de 0,05, ou seja, são hipóteses fortes com base na amostra coletada.

Quanto ao design do chatbot, estudos apontam como fatores cruciais: personalizar o tratamento ao usuário (Jain et al., 2018), incluir conversa fiada (Kerly et al., 2007; Jain et al., 2018), manter o foco do usuário (permitir digressões, porém retornar ao tópico inicial) (Kerly et al., 2007), acompanhar o contexto da conversa (Brennan, 1991; Jain et al., 2018) e ser claro quanto às capacidades do robô (Jain et al., 2018).

Quanto ao aprendizado proporcionado pelo chatbot e à satisfação com a experiência, um fator recorrente na literatura é a atitude dos usuários (Arbaugh, 2001; Söllner et al., 2018). Quão mais positiva, ou seja, quão mais predispostos a interagir com o chatbot e mais habilidosos com tecnologia, melhores as notas obtidas em testes e concedidas ao aprendizado.

4.3 Revisão sistemática de literatura

Winkler & Söllner (2018) realizaram uma revisão sistemática de literatura para avaliar o impacto do uso de chatbots na educação. Dentre 1405 artigos sobre chatbots nos campos de gestão, educação, sistemas de informação e psicologia, com base nos abstratos e títulos foram escolhidos e examinados individualmente 80.

Um fator crucial do emprego de chatbots na educação, segundo os autores, é que eles permitem os alunos ativamente controlar seu aprendizado, que é “um principal fator de sucesso para o aprendizado efetivo segundo a teoria de aprendizado construtivista predominante” (Winkler & Söllner, 2018). Além disso, chatbots podem, com pouco investimento, compensar a falta de apoio individual a estudantes, enquanto contratar mais professores e ajudantes nem sempre é possível por falta de recursos.

Quanto à metodologia, os pesquisadores concluíram que uma maneira comum de se medir o aprendizado afetivo é através da satisfação dos estudantes, sendo esse modo de aprendizado relacionado à motivação intrínseca dos alunos, ou seja, o desejo de participar mais em certas atividades que outras.

Outro achado importante é que o sucesso de um chatbot é muito dependente da sua acessibilidade, ou seja, integração com sistemas que os alunos já utilizam no dia a dia fora do ambiente de ensino, como Facebook e WhatsApp.

O estudo concluiu que chatbots, por criarem experiências individuais de aprendizado para alunos, melhoram seu aprendizado e reduzem a carga no sistema educacional, especialmente em cursos online massivos e grandes salas de aula, onde apoio individualizado é impraticável devido a restrições financeiras.

4.4 Comparação de frameworks

Há diversos *frameworks* para o desenvolvimento de chatbots no mercado: Dialogflow, do Google; RASA, solução de código aberto; LUIS, da Microsoft; etc. Em um recente estudo (Qi et al., 2020), os *frameworks* mais proeminentes foram avaliados na capacidade de detectar intenção, um fator chave para a construção de chatbots e que varia muito com a implementação. Diferentes algoritmos de aprendizado de máquinas são utilizados por cada ferramenta, levando a variações significativas na performance. O estudo encontrou que, na média entre 3 *datasets* (conjuntos de dados), o Watson ficou em primeiro lugar na acurácia de detecção de intenção entre os 6 robôs participantes. Além disso, ele teve o menor tempo de treinamento que os concorrentes por um fator de no mínimo 7 vezes até mais de 200 vezes.

Considerando esses aspectos não-funcionais e outros como interface e funções disponíveis (descritas a seguir), o Watson foi escolhido como base para desenvolver o trabalho.

4.5 Watson Assistant

O Watson Assistant é uma das ramificações do Watson, conjunto de ferramentas de inteligência artificial da IBM. Ele permite desenvolver chatbots através de uma interface gráfica na web, pode receber e produzir textos e áudios e possui diversas integrações com outras plataformas.

Features (características, funções ou recursos) do Watson Assistant (IBM, [s.d.]):

- **Habilidades:** podem ser de 3 tipos – diálogo (interagir com o usuário, receber e enviar mensagens), pesquisa (procurar informações em bancos de dados ou na internet, podendo atualizar com frequência seu conteúdo) e ação (similar a diálogo, porém mais focada nos objetivos do usuário e menos complicada para desenvolver).
- **Agentes:** possuem um campo de expertise, no qual são preparados para responder perguntas e executar ações. Cada agente pode possuir uma habilidade de cada tipo, totalizando 3.
- **Nós:** a habilidade de diálogo define o rumo da conversa como uma árvore lógica. Nós podem ter filhos, criando um diálogo linear. Cada nó é ativado quando suas condições são atendidas. Elas podem ser simplesmente uma intenção, assim como combinações lógicas de contextos, entidades e intenções. Ao ser ativado, o nó pode guardar um contexto, responder com texto, imagem ou opções, passar a conversa para um agente humano etc. Por último, ele pode aguardar uma resposta do usuário, pular para outro nó ou pular para seus filhos.
- **Intenções:** possuem um nome único e textos de treinamento, com base nos quais o algoritmo de aprendizado de máquina as identificará no texto de entrada do usuário.
- **Entidades:** o desenvolvedor pode criar entidades próprias, por exemplo para reconhecer acrônimos. Também há entidades predefinidas que auxiliam na identificação de datas (ex.: sexta-feira), horários (ex.: às 10), porcentagens, valores monetários e números. Entidades têm valores, cada um dos quais pode ter sinônimos.
- **Variáveis de contexto:** nós do diálogo podem armazenar informações sobre a conversa através de variáveis de contexto, assim como procurar entre as armazenadas como condição de ativação.

- Integração: o chatbot pode realizar comunicação com serviços externos através de Webhooks. Cada nó do diálogo é capaz de ativar uma webhook ou uma habilidade de ação. Também é possível fazer o robô operar através de plataformas diferentes, como um site próprio, WhatsApp, Slack, Facebook Messenger etc.
- Digressões: no meio de um diálogo que tem um objetivo, um usuário pode repentinamente mudar de assunto para tratar de outro objetivo. Caso a digressão seja habilitada para um nó, é possível voltar ao ponto original da conversa antes da mudança de assunto.
- Respostas múltiplas: um nó pode responder diferentemente dependendo de condições, como contextos e entidades. Isso permite simplificar a árvore de diálogo, agrupando respostas similares em apenas um nó, ao invés de criar vários nós com diferentes condições de ativação.
- Slots: um nó pode possuir requisitos de informações necessárias para completar seu objetivo. Por exemplo, um nó de reserva de mesa em um restaurante pode necessitar da data, horário e quantidade de pessoas. É possível que o usuário apresente todas essas informações em apenas uma declaração, assim como ele pode fazer o pedido inicial sem nenhuma especificação. Nesse caso, o chatbot realizará perguntas atreladas a cada uma das informações necessárias até obtê-las do usuário, por exemplo “Que horas você quer reservar?” caso o horário não tenha sido informado.

MÉTODOS DE AVALIAÇÃO DE RESULTADOS

5 Métodos de avaliação de resultados

Como deve ser avaliada a qualidade de um chatbot? Essa pergunta pode parecer simples de início, mas, na verdade, envolve diversos fatores. O primeiro critério que vem à mente é a cobertura, ou seja, a proporção entre as vezes que o robô entendeu e não entendeu o texto enviado pelo usuário. Porém, esse valor não diz toda a realidade. Ele ofusca, por exemplo, se o robô entendeu corretamente o que o usuário queria. De fato, esse é a informação que se deseja saber, porém não há jeito fácil de obtê-la no caso de chatbots orientados a diálogo. Caso existisse uma lógica infalível para determinar a intenção do usuário, ela teria sido utilizada em primeiro lugar. Ainda existe a possibilidade de avaliar manualmente a conversa e determinar, para cada mensagem, se a resposta obtida é adequada. Essa solução, porém, não escala bem com a quantidade de mensagens trocadas. No caso de chatbots orientados a tarefas, para determinar a taxa de sucesso basta fazer uma pergunta ao usuário ao fim da sessão: se ele atingiu o seu objetivo.

Então, para complementar a informação da cobertura, o que se utiliza com mais frequência é a avaliação pelo próprio usuário, conforme visto nos estudos anteriores. Apesar de a avaliação ser subjetiva, com uma quantidade grande de usuários ela se torna um parâmetro relevante. A avaliação normalmente ocorre na forma de um questionário, onde eles atribuem uma nota para cada critério.

Ainda há outros aspectos a se considerar. Assumindo que o robô entendeu o que o usuário disse e que esse entendimento corresponde ao que o usuário queria de fato dizer, ainda falta saber se a resposta foi satisfatória. Assim como o fator anterior, não existe uma maneira clara de medir isso para cada pergunta realizada, então esse tipo de métrica também é obtido, normalmente, através de questionário (Casas et al., 2020).

Mais específico ao âmbito desse trabalho, é importante coletar informações como: o usuário gostou da experiência? Ele achou melhor que ler um texto ou ver um vídeo? Ele sente que aprendeu? Todas essas métricas são melhor obtidas através de avaliações pelos usuários e servem para verificar o potencial de um chatbot como uma ferramenta educativa.

No geral, não existe ainda uma maneira padronizada no meio científico para avaliar a qualidade de um chatbot (Casas et al., 2020). No estudo conduzido pelos pesquisadores, examinou-se 2 publicações sobre métodos de avaliação e outras 53 sobre chatbots, que continham algum método de avaliação. O objetivo era examinar quais eram as técnicas mais populares no meio acadêmico para avaliação de qualidade de chatbots. Eles concluíram que, conforme encontrado em um artigo anterior (Radziwill & Benton, 2017), os métodos costumam se alinhar

com o conceito de usabilidade de software presente na norma ISO 9241: “A **efetividade, eficiência e satisfação** com as quais usuários alcançam objetivos específicos em ambientes particulares” (International Standards Organization, 2018, tradução própria). Segundo os autores, esses 3 conceitos podem ser aplicados da seguinte maneira no contexto de chatbots:

- **Efetividade:** capacidade de interpretar comandos precisamente, facilidade de uso, acurácia linguística das saídas, transparência sobre ser um chatbot, inclusão de erros para aumentar realismo
- **Eficiência:** capacidade de atingir o objetivo, sendo robusto ao tratar entradas inesperadas, *graceful degradation* (falhar graciosamente – providenciar uma solução alternativa em caso de falha ou deixar o usuário ciente das limitações)
- **Satisfação:** dar boas-vindas, demonstrar personalidade, dar dicas de conversa, providenciar informações de emoção através de tom, inflexão e expressividade, ser autêntico e caloroso, deixar tarefas mais interessantes e divertidas, possuir informações confiáveis

Entre as 53 publicações analisadas, 35,85% usaram um método de avaliação relacionado a efetividade, outros 58,49%, um relacionado a eficiência e 52,83%, um relacionado a satisfação. 47,17% usaram métodos de pelo menos 2 tipos (por exemplo, um de eficiência e outro de satisfação, a combinação mais comum), 52,83% usaram de apenas 1 tipo e nenhum usou os 3 ao mesmo tempo. Enfim, concluiu-se que não há um conceito que prevaleça em relação aos outros, especialmente considerando a variação ao longo do período de 5 anos analisado. Além do mais, observa-se a tendência de crescimento da adoção de critérios de eficiência. Por último, alguns métodos são mais adotados para domínios específicos, por exemplo, chatbots baseados em tarefas (como aqueles que reservam mesas de restaurante ou quartos de hotel) costumam usar mais medidas de eficiência, enquanto aqueles no domínio de diálogo (como o do presente trabalho) costumam usar medidas de satisfação, já que o foco não é o cumprimento de tarefas, e sim a própria discussão. Isso concorda com o encontrado na Seção 4.2, que avalia mais especificamente o contexto de chatbots na educação.

METODOLOGIA

6 Metodologia

Essa seção trata dos requisitos do trabalho, como eles serão medidos e, em seguida, dos passos tomados para realizá-lo.

6.1 Requisitos

O projeto inicia com a definição dos requisitos. Por se tratar de um projeto de software, os requisitos podem ser classificados entre funcionais e não-funcionais.

6.1.1 Requisitos não-funcionais

Requisitos não-funcionais do desenvolvimento de um chatbot podem ser fatores como taxa de acerto de intenção, tempo de resposta e taxa de cobertura.

Do ponto de vista do usuário, ficar aguardando longos períodos por uma resposta pode rapidamente causar o desinteresse, então espera-se que o tempo de resposta médio não supere 5 segundos, uma duração bem aceita (Gnewuch et al., 2018).

Outros aspectos não-funcionais não serão considerados requisitos. Por exemplo, uma determinada taxa de acerto de intenção não pode ser medida diretamente (conforme coberto na Seção 5) e não tem um impacto direto na qualidade da experiência ou do aprendizado do usuário, sendo essas características resultantes de uma combinação de fatores.

6.1.2 Requisitos funcionais

Conforme definido na Seção 2, espera-se que o robô seja, em primeiro lugar, capaz de responder perguntas simples sobre o tema, cobrindo aspectos básicos como recursos vivos, minerais e energéticos. Além disso, ele deve educar e fomentar o interesse dos usuários no tema. Em termos de requisitos, temos, então:

1. abranger perguntas sobre a AAz em si (definição, origem, extensão etc.) e os temas de petróleo e gás natural, outros recursos energéticos, pesca e biologia marinha;
2. providenciar uma experiência positiva aos usuários;
3. contribuir para a educação dos usuários;

Em seguida, conforme o encontrado na Seção 4.2, temos:

4. acompanhar o contexto da conversa;
5. dar respostas curtas;

6. permitir o usuário guiar a conversa;
7. ser de fácil acesso.

Para auxiliar na determinação do cumprimento dos requisitos 2 e 3, serão coletadas (através de formulário) as seguintes métricas, que se encaixam em métodos de avaliação de Satisfação (S) ou Efetividade (E) (vide a Seção 0):

Em uma escala de 0 (nem um pouco) a 5 (muito)...

1. ... com que frequência o robô entendeu a sua pergunta? (E)
2. ... quanto você considera que aprendeu com a experiência? (S)
3. ... qual é seu nível de satisfação com a experiência? (S)
4. ... quanto você considera as respostas confiáveis? (S)
5. ... quanto a experiência atendeu às suas expectativas? (S)
6. ... quanto você considerou essa experiência melhor que assistir um vídeo ou ler um texto? (S)
7. ... quanto você recomendaria esse chatbot a amigos? (S)

A escolha da pontuação de 0 a 5 é baseada em uma Escala de Likert (Likert, 1932), sendo unipolar positiva de 5 pontos. O uso mais tradicional é o de uma escala bipolar (que vai de uma pontuação negativa para uma positiva), porém pesquisas mostram desvantagens em comparação à escala unipolar (apenas positiva ou apenas negativa), como: maior esforço cognitivo para avaliar; possibilidade de confusão quanto ao significado do ponto central (significa neutro, “não sei” ou “algumas coisas positivas, outras negativas”?); escalas bipolares bem formadas são mais longas e requerem mais leitura (Market Research Institute International, [s.d.]). Além disso, escalas bipolares produzem o Efeito de Aquiescência (Saris et al., 2010), quando respondentes tendem a selecionar que concordam com afirmações por uma mistura de fatores, entre eles por um sentimento de respeito aos pesquisadores, por esperar que os pesquisadores concordem com as afirmações e delegar a decisão a eles ou por precisar de menos esforço para concordar com algo que para cuidadosamente escolher entre cada nível de concordância ou discordância.

A escolha de uma escala positiva possui um encaixe melhor com o teor das perguntas realizadas, por exemplo: as respostas para as perguntas 1 a 5 são necessariamente positivas, já que não é possível, respectivamente:

1. o chatbot entender a pergunta do usuário com uma frequência negativa;
2. o usuário “desaprender” com a experiência;
3. o usuário estar negativamente satisfeito – normalmente, qualquer nível de insatisfação é denotado pela nota 0 na escala;

4. o usuário achar as respostas menos que nem um pouco confiáveis;
5. o usuário ter suas expectativas menos que nem um pouco atendidas.

As respostas 6 e 7 podem assumir valores negativos, porém, nesse caso, assume-se que é improvável que um usuário atribua uma nota negativa. Por exemplo: é possível, teoricamente, recomendar negativamente um produto, porém esse tipo de resposta é improvável no contexto desse trabalho, por se tratar de uma ferramenta gratuita, de pequena escala e que não atende uma necessidade existente das pessoas. Ou seja, caso um usuário fique descontente com a experiência, não faz sentido para ele compartilhá-la com outras pessoas apenas para dizer que não é boa, já que elas sequer teriam entrado em contato com o trabalho de outra forma.

As respostas serão analisadas e espera-se que cada uma atinja uma média acima de 2,5, o equivalente a um ponto neutro, assim como a média entre todas as métricas. Vale ressaltar a diferença entre o ponto neutro da escala unipolar positiva e o ponto central da escala bipolar: o ponto neutro fica entre algo um pouco positivo e muito positivo, enquanto o ponto central não é nem positivo nem negativo. Dado o objetivo central do trabalho, de desenvolver uma ferramenta capaz de educar e instigar a curiosidade do público em relação à AAz, é essencial que os usuários gostem do chatbot e da experiência. Por isso, não basta uma nota positiva e neutra, ela precisa ser, no mínimo, melhor que isso.

6.2 Desenvolvimento do chatbot

O primeiro passo no desenvolvimento é a obtenção de perguntas. Para obter uma grande quantidade de perguntas que representam as curiosidades do público geral, analisou-se o *dataset* Question Answering V2.1 do MS MARCO, da Microsoft. Ele consiste em um conjunto de mais de 1,000,000 perguntas feitas no motor de busca Bing entre os anos de 2016 e 2020 e as respectivas respostas, entre outras informações. O foco original do Question Answering é providenciar um corpus para treinar e testar algoritmos que encontram a resposta de uma busca, porém, no âmbito desse trabalho, ele é utilizado como fonte direta de perguntas para o chatbot e indireta de inspiração para a formulação de novas perguntas. Observação: o *dataset* possui perguntas apenas em inglês. Exemplos a seguir são dados em português, mas toda interação com ele foi feita em inglês na realidade e os resultados foram traduzidos.

Uma vez obtido o conjunto de dados a ser usado pelo chatbot, ele precisa ser programado. É necessário transformar uma planilha de perguntas e respostas em uma série de intenções, nós de diálogo e, opcionalmente, entidades e variáveis de contexto. A solução trivial é criar uma intenção para cada pergunta e um nó de diálogo para cada intenção contendo a

resposta. Porém, ela apresenta um grave problema: a pergunta só será identificada se o usuário digitar algo muito similar ao que está na planilha. Por exemplo, se o usuário perguntar “Qual é a definição de tartaruga?” ao invés de “O que é tartaruga?”, que é o encontrado na planilha, o chatbot não entenderá. A solução para isso é adicionar elocuções, uma *feature* existente que permite identificar a mesma pergunta de diferentes maneiras. Porém, o trabalho de pensar todas as elocuções possíveis para cada pergunta é sujeito a erros e não escala bem com o tamanho do conjunto de dados. Esse problema é agravado pela variação de forma na linguagem natural, especialmente na língua portuguesa, o que gera ainda mais maneiras de redigir perguntas, conforme exemplificado na Seção 4.1.4. A solução adotada envolve classificar as perguntas para permitir detectar elocuções de forma automática, o que será detalhado na Seção 6.2.2. Então, tornou-se necessário um algoritmo de determinação de intenção que considerasse essas classificações, que é descrito na Seção 6.2.3.

Por último, é necessário implementar o algoritmo e passar as informações para o chatbot. Há duas maneiras de se realizar essa etapa: uma delas é através da interface web do Watson Assistant, onde é necessário clicar, digitar, copiar e colar textos; outra é escrever um arquivo JSON de acordo com as especificações da API do Watson e realizar upload no site. JSON é um padrão de indústria para transferência de dados padronizados. Um arquivo aceito pela API para representar o chatbot está exemplificado na Figura 1. Nele, os valores das chaves “intents”, “entities” e “dialog_nodes” são listas de dicionários; os das chaves “metadata” e “system_settings” são dicionários; e os das chaves restantes são *strings* ou booleanos. Pode-se realizar tanto o upload quanto o download do arquivo, ou seja, é possível pegar o chatbot existente, convertê-lo a esse formato, editar o arquivo e devolver a versão nova ao site.

Figura 1 – Arquivo JSON que contém as informações do chatbot, com alguns valores oclusos

```
{
  "intents": [],
  "entities": [],
  "metadata": {},
  "dialog_nodes": [],
  "counterexamples": [],
  "system_settings": {
    "disambiguation": {
      "prompt": "Você quis dizer:",
      "enabled": false,
      "randomize": true,
      "max_suggestions": 3,
      "suggestion_text_policy": "title",
      "none_of_the_above_prompt": "Nenhuma das anteriores"
    },
    "human_agent_assist": {
      "prompt": "Você quis dizer:"
    }
  },
  "learning_opt_out": false,
  "name": "Amazônia Azul",
  "language": "pt-br",
  "description": ""
}
```

Fonte: Autor

A existência de uma API possibilita automatizar a criação do robô, o que agiliza a tarefa e reduz a probabilidade de erros. Para isso, foi feito um código em Python para gerar o arquivo JSON.

Ambos os códigos produzidos nesse capítulo estão disponíveis na íntegra em <https://github.com/C4AI/blab-filter-msmarco>.

6.2.1 Obtenção de perguntas

Foi utilizado o conjunto de *training* do Question Answering V2.1, composto de mais de 800 mil perguntas de tópicos diversos. Para encontrar perguntas relevantes ao tema da AAZ, é necessário filtrá-las, uma tarefa notoriamente complexa. Esse desafio é conhecido como *Topic Modelling*.

A solução trivial é avaliar as perguntas uma por vez, o que resulta em um conjunto final de qualidade garantida, porém demanda tempo em excesso.

Uma solução mais elaborada é procurar por palavras ou padrões entre as perguntas, por exemplo: manter todas as perguntas que possuem as palavras “Amazônia Azul”, descartar todas as que possuem “matemática”. Para fins similares, são amplamente utilizadas expressões regulares (também conhecidas por RegEx), uma notação que permite estabelecer padrões de palavras para realizar buscas. A utilização de expressões regulares para filtrar um conjunto de dados é um método simples e fácil de implementar, porém gera muitos falsos positivos e

negativos, levando a um resultado de baixa qualidade. É possível melhorar seus resultados, porém isso aumenta a complexidade exponencialmente.

A abordagem empregada se baseou em filtrar o conjunto inicial em passos, indo do nível mais grosseiro para o mais granular. Primeiro, perguntas foram agrupadas por similaridade de conteúdo. Em seguida, agrupamentos foram avaliados a partir das palavras que melhor os representavam para selecionar os relevantes. Por último, as perguntas pertencentes aos agrupamentos escolhidos foram avaliadas manualmente e com auxílio de RegEx. O código está disponível no repositório do trabalho, distribuído entre os arquivos “Sentence Transformers.ipynb”, “choose_questions.py”, “extract_questions.py” e “utils.py”.

O passo-a-passo inicia com a utilização do SBERT, um *framework* de Python para o cálculo de *embeddings* de frases e texto (Grootendorst, 2020; Reimers & Gurevych, 2020). O cálculo de *embeddings* das perguntas resultou em vetores com 768 componentes cada, o que representa muito bem cada pergunta, porém é informação demais para se processar. Então, antes de prosseguir, foi realizada uma redução de dimensionalidade através da biblioteca *umap* do Python, aproximando os 768 componentes para apenas 5 enquanto mantendo a representatividade das perguntas. Com as perguntas mapeadas, elas foram agrupadas em *clusters* através do método de similaridade por cosseno utilizando a biblioteca *hdbscan* do Python, o que resultou em cerca de 5000 *clusters*.

Para determinar quais *clusters* eram relevantes, foram calculadas as 10 palavras mais representativas de cada um através de uma implementação de TF-IDF. Então, todos *clusters* contendo as palavras “marinho”, “oceano”, “mar”, “petróleo”, “praia”, “corrente”, “maré” ou “onda” foram selecionados e visualizou-se as 10 palavras mais representativas de cada. Manualmente se anotou o identificador dos que pareciam relevantes, formando uma lista. Todas as perguntas pertencentes a *clusters* dessa lista foram selecionadas, totalizando 1844. Dentre essas, algumas ainda estavam completamente fora do tema, por exemplo todas relacionadas a correntes elétricas (ao invés de correntes oceânicas, que é o agrupamento que as capturou), então elas foram retiradas pelo uso de expressões regulares e palavras-chave. Assim, restaram 570, incluindo algumas repetições.

O último passo foi analisar as perguntas uma a uma. Nesse momento, muitas foram descartadas e as restantes foram categorizadas com um ou mais dos seguintes rótulos: “tartarugas”, “corais”, “energia de maré”, “petróleo”, “gás”, “física”, “oceano”, “onda”, “maré”, “corrente”, “flora”, “fauna”, “oceanografia”, “turismo”, “engenharia de petróleo”, “saúde”, “litoral”, “proteção ambiental”, “geologia” e “outras”. A adição de rótulos dificulta a tarefa de

seleção, porém é necessária para etapas futuras, conforme será explicado. Ao final, restaram 169 perguntas.

Ao preparar as respostas para as perguntas, percebeu-se que algumas ficavam muito extensas devido à necessidade de explicar mais de um conceito. Adotou-se, então, a estratégia de subdividir as perguntas, tornando as respostas mais granulares. Isso vai de encontro à diretriz encontrada na literatura de permitir o usuário controlar a experiência, pois incentiva que ele faça mais perguntas na direção da sua curiosidade, ao invés de receber muita informação de uma vez. Um exemplo: para a pergunta “O que é um plâncton?”, a resposta original era: “Plânctons são um grupo de organismos com pouca capacidade de locomoção e que são transportados por correntes d'água. (...) Eles podem ser divididos entre fitoplânctons (que produzem a própria energia através de fotossíntese) e zooplânctons (que se alimentam de outros seres vivos).” A pergunta foi então dividida, criando mais duas: “O que são fitoplânctons?” e “O que são zooplânctons?”, permitindo tanto reduzir a resposta original quanto incorporar informações extras nas outras respostas, que ficariam excessivas na primeira.

Toda informação obtida nessa seção foi salva em uma planilha, onde cada linha contém uma pergunta, sua resposta e seus rótulos.

6.2.2 Classificação das perguntas

O Watson é especializado em detectar intenções diretamente, ou seja, comparando o texto do usuário com o texto de exemplo (e elocuções) de cada pergunta. Porém, adicionar todas as elocuções possíveis para cada pergunta é uma tarefa árdua e ineficiente. Então, com o objetivo de considerar de forma automática as elocuções de cada pergunta, elaborou-se a metodologia descrita a seguir.

Através do uso elaborado de entidades e variáveis de contexto, pode-se detectar a intenção de uma pergunta através de suas partes, em adição à detecção direta da pergunta completa. De forma arbitrária, definiu-se que cada pergunta é composta pelas seguintes partes: Contexto, Modificador, Substantivo e Recipiente. Essas categorias não possuem relação direta com seus equivalentes morfosintáticos. Elas foram adicionadas na forma de colunas na planilha de perguntas e preenchidas de tal forma que cada pergunta é representada por uma combinação única desses 4 conceitos.

Antes de entrar no significado de cada parte, veja um exemplo de aplicação: a palavra “por que” é muitas vezes escrita “pq” na linguagem informal. Assim, para cada pergunta que a contém tipo seria necessário adicionar uma elocução em que “por que” é substituído por “pq”.

Ao invés disso, essa palavra foi configurada como uma entidade com diversas grafias, como “porque”, “porquê”, “por quê” e “pq”. Tecnicamente, essas palavras possuem significados diferentes, mas, na prática, elas são usadas de maneira intercambiável na linguagem informal. Todas as perguntas contendo essa palavra foram associadas a essa entidade, de modo que qualquer grafia anotada pode ativar a detecção da intenção. Caso fossem geradas as elocuções de cada intenção individualmente com base nesses critérios, cada pergunta do tipo “por que” teria 4 elocuções apenas devido a essa palavra. Se a pergunta tivesse outras palavras com o mesmo tratamento, o efeito delas seria multiplicativo, o que acarretaria a geração de até mesmo dezenas de elocuções por pergunta, o que é extremamente ineficiente em termos de memória e pode afetar negativamente a performance do chatbot. Por exemplo, a pergunta “O que são actinopterígeos?” possui duas partes que podem ser escritas de diferentes maneiras: “o que são” pode ser “o que é”, “definição de”, “defina” ou “significado de” (5 formas) e “actinopterígeos” pode ser “Actinopterygii” (2 formas). Logo, seriam necessárias 10 elocuções para contemplar todas as combinações possíveis das 2 partes da pergunta. Ao invés de gerar todas as elocuções, a divisão da pergunta em partes permite detectá-las automaticamente simplesmente definindo quais são suas partes e quais são as diferentes formas de escrever cada uma.

A palavra “por que”, no exemplo anterior, define o tipo de questão. Portanto, na classificação elaborada, ela entra na categoria de Modificador. Existe um conjunto limitado de modificadores possíveis, sendo eles: “causa”, “definição”, “diferença”, “é”, “efeito”, “existe”, “explicar”, “listar”, “maior”, “maiores”, “motivação”, “por que”, “quantidade” e “responsável”. Alguns exemplos de perguntas e respectivos modificadores estão no Quadro 4. Cada Modificador recebe sinônimos, que permitem detectar aquele tipo de pergunta de diferentes maneiras, conforme o apresentado no Quadro 5.

Quadro 4 – Pares pergunta/modificador

Pergunta	Modificador
O que é um coral?	definição
Petróleo é um recurso renovável?	é
Por que tartarugas estão em extinção?	por que
Quem está causando a acidificação dos oceanos?	responsável

Fonte: Autor

Quadro 5 – Pares modificador/sinônimos

Modificador	Sinônimos
definição	significa, significado, defina, definir, o que é, o que são
efeito	consequência, impacto
por que	porque, porquê, por quê, pq
responsável	quem

Fonte: Autor

O Contexto é o mesmo que o rótulo (definido na seção anterior), com exceção de alguns rótulos que não definem contexto, como “fauna”, “flora” e “outras”. Idealmente, esses rótulos não deveriam existir, pois cada pergunta deveria ter ao menos um contexto. Isso foi feito para agrupar perguntas de temas distintos, porém pequenos. Por exemplo, “O que é um filo?” é uma pergunta que encaixaria no Contexto “filo”, porém, ela seria a única, então resolveu-se incluí-la sob o rótulo genérico de “fauna”. Criar um Contexto para uma única pergunta é possível, mas não apresenta nenhum benefício, já que o Contexto é utilizado exclusivamente para possibilitar fazer perguntas seguidas dentro de um tema.

Algumas perguntas são perfeitamente sintetizadas apenas pela combinação de Contexto e Modificador. Já outras precisam ainda de mais informação, então utiliza-se o Substantivo e/ou o Recipiente. Diferentemente do Modificador, esses conceitos não possuem uma única definição para todas as perguntas, podendo assumir diferentes papéis na frase dependendo do Modificador e da presença um do outro. Ou seja, para cada Modificador, existem até 4 combinações possíveis: com apenas Substantivo, com apenas Recipiente, sem nenhum dos dois ou com ambos. Essa distinção é necessária, pois certos tipos de pergunta podem apresentar o Contexto como sujeito, como predicado ou nem o incluir. Por exemplo, perguntas do tipo “definição” são sobre o significado do Recipiente. Se não houver recipiente, elas se referem ao Contexto. Em ambos os casos, o Substantivo aparece antes do alvo para torná-lo mais específico. Veja exemplos no Quadro 6.

Quadro 6 – Exemplos de Substantivo e Recipiente para perguntas do tipo “definição”

Pergunta	Rótulo	Substantivo	Recipiente	Alvo da pergunta
O que é um coral?	corais			corais
O que é um recife de coral?	corais	recife		recife de corais
O que é petróleo Pré-Sal?	petróleo		pré-sal	petróleo pré-sal
O que é Albatroz?	fauna		albatroz	albatroz
O que é Projeto Albatroz?	outras	projeto	albatroz	projeto albatroz

Fonte: Autor

Cada pergunta é associada a uma intenção única no chatbot. Para formá-la, concatenam-se o conteúdo das 4 colunas na ordem apresentada. Por exemplo: a pergunta “O que é petróleo Pré-Sal?” possui Contexto: “petróleo”; Modificador: “definição”; Substantivo: nenhum, Recipiente: “Pré-Sal”, logo, a intenção fica: “petróleo—definição-pré-sal”. Outros exemplos de perguntas e respectivas intenções estão no Quadro 7.

Quadro 7 – Pares pergunta/intenção

Pergunta	Intenção
O que são recifes artificiais?	corais--definição-recife-artificial
Quais são os prós e contras da energia de maré?	energia_de_maré--listar-prós-e-contras
Que estados produzem mais petróleo?	petróleo--maiores-estado
Qual é a diferença entre uma tartaruga e um jabuti?	tartarugas--diferença-jabuti
Como é o ciclo de vida de uma tartaruga?	tartarugas--definir-ciclo-de-vida

Fonte: Autor

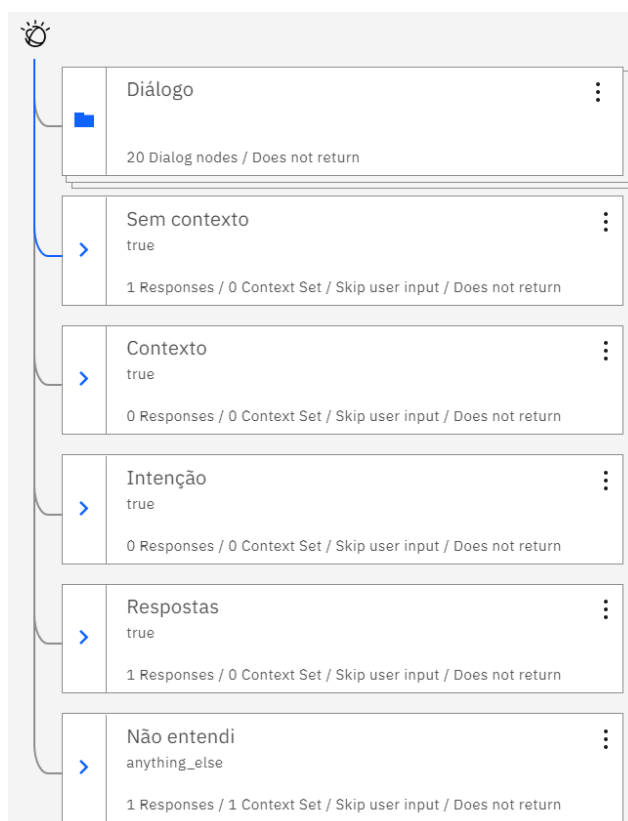
A tarefa de adicionar as informações de Modificador, Substantivo e Recipiente a cada pergunta pode parecer ineficiente de início. Porém, uma vez definidos os modificadores possíveis e qual a lógica deles, ela é relativamente rápida e traz o benefício da geração automática de elocuições, que não ocorre facilmente de outra maneira. Além disso, após popular dados suficientes com essas informações, é possível treinar um modelo de aprendizado de máquina para rotular novos dados de forma automática. Dado o escopo do presente trabalho, essa possibilidade não foi investigada mais a fundo.

6.2.3 Arquitetura dos nós de diálogo

Uma vez classificadas as perguntas, é necessário traduzir os conceitos criados em *features* do Watson, ou seja, entidades e variáveis de contexto. Para isso, desenvolveu-se um algoritmo, que será explicado a seguir. Ele depende da seguinte arquitetura:

- Utiliza-se uma variável de contexto chamada “contexto”, que guarda o tema da última pergunta feita pelo usuário.
- Rótulos, Modificadores, Substantivos e Recipientes são programados na forma de entidades, como pode ser visto na Figura 3.
- O Watson trata os nós de diálogo como uma árvore. Sempre que ele recebe uma mensagem do usuário, ele busca como responder começando pela raiz (parte de cima na interface) e descendo até encontrar o primeiro nó cujas condições de ativação são satisfeitas. Ele executa o nó e pode tomar uma de 3 ações em seguida: esperar uma nova entrada do usuário, saltar para um outro nó da árvore ou começar imediatamente um novo ciclo de busca nos filhos desse nó. Quando ele salta para outro nó, ele pode executá-lo imediatamente ou aguardar uma nova entrada para começar a próxima busca a partir desse ponto.
- Os nós são agrupados em pastas propriamente ditas ou como filhos de um nó que se comporta como uma pasta, conforme a Figura 2. As pastas são:
 - Diálogo: contém nós que não estão presentes na planilha, como os de introdução, ajuda, contato, informações sobre o trabalho etc.
 - Sem contexto: contém nós relativos a perguntas sem Contexto, que ativam apenas com base em Modificador, Substantivo e Recipiente.
 - Contexto: contém nós que detectam um Contexto na pergunta e alteram a variável de contexto para o valor detectado.
 - Intenção: contém subpastas, uma para cada contexto possível, que são ativadas pela variável de contexto. Dentro de cada uma, há nós que identificam a intenção com base em Modificador, Substantivo e Recipiente.
 - Respostas: contém nós com as respostas.
 - Não entendi: contém nós que ativam caso todo o resto falhe, ou seja, nenhuma intenção seja detectada.

Figura 2 – Estrutura de nós de diálogo



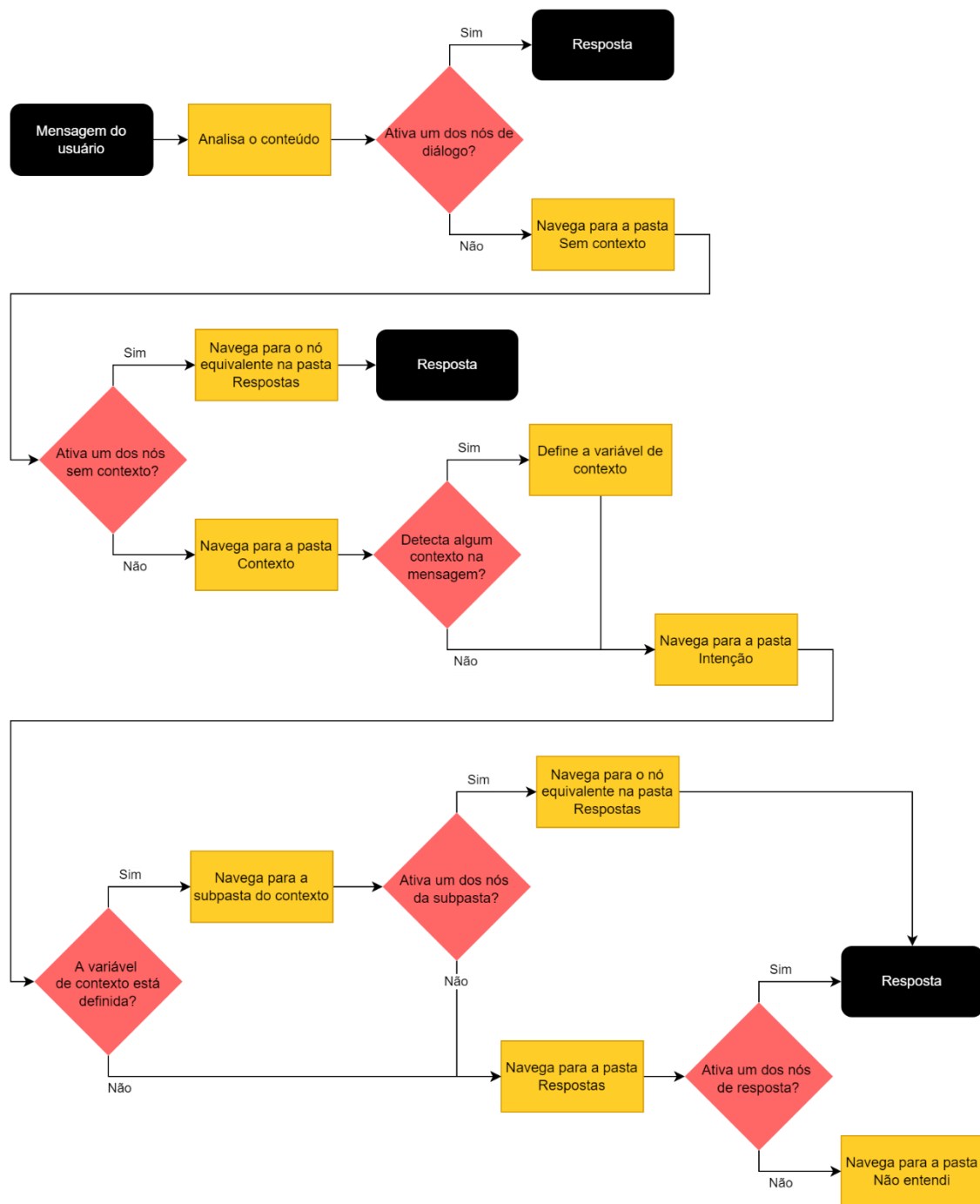
Fonte: Autor

Figura 3 – Entidades

<input type="checkbox"/> Entity (6) ↑	Values
<input type="checkbox"/> @bool	bem, sim, não, mal
<input type="checkbox"/> @modificador	causa, onde, porque, quantidade, responsável, é, composição, definição, detalhar, diferença, efeito, existe, explicar, listar, maior, mai...
<input type="checkbox"/> @pronomes	seu, você
<input type="checkbox"/> @recipiente	acidificação, actinoptérgeo, albatroz, alga, altura, ambiente, anêmona, aquicultura, baleia jubarte, bioma, branqueamento, brasil, c...
<input type="checkbox"/> @rótulos	fauna, correntes oceânicas, corais, baleias_turismo, animais marinhos, baleias, amazônia azul, flora, energia de maré, corais_turism...
<input type="checkbox"/> @substantivo	prós, animal marinho, baleia, extinção, geração, animais marinhos, variação, turismo, tubarão, tipo, tempo entre altas, tamanho, sím...

Fonte: Autor

Figura 4 – Algoritmo dos nós de diálogo em forma de fluxograma



Fonte: Autor

O algoritmo serve para determinar qual caminho lógico será seguido até que se determine a intenção desejada do usuário sempre que ele mandar uma mensagem. Ele está ilustrado na forma de fluxograma na Figura 4, onde caixas pretas representam entradas e saídas, caixas amarelas representam ações e losangos representam decisões. Sempre que ele chega a uma resposta, ele se encerra. A complexidade se deve ao fato de acomodar tanto a possibilidade de

detectar perguntas através de suas partes, conforme definido na seção anterior, assim como diretamente pelo exemplo principal, que é o método padrão do Watson.

Além da representação em diagrama, ele está descrito em forma de texto a seguir:

1. o chatbot recebe uma pergunta do usuário;
2. analisa o conteúdo e obtém a intenção mais provável e entidades presentes (rótulo, Modificador, Substantivo, Recipiente);
3. verifica se a intenção mais provável ativa algum dos nós na pasta de Diálogo, caso sim, retorna a resposta;
4. navega para a pasta Sem contexto e verifica se algum nó é ativado pelas entidades presentes na pergunta, caso sim, redireciona para o nó com a resposta;
5. navega para a pasta de Contexto e verifica a presença de algum rótulo contextual na pergunta, caso sim, redefine o valor da variável de contexto para o que foi encontrado;
6. navega para a pasta de Intenção e verifica se a variável de contexto está definida, caso sim, realiza o passo 6.a;
 - a. entra na subpasta do contexto atual e verifica se, sob aquele Contexto, a combinação de entidades ativa um dos nós, caso sim, redireciona para o nó com a resposta;
7. navega para a pasta de Respostas e verifica se a intenção mais provável possui uma probabilidade maior que 70%, caso sim, ativa o nó de resposta;
8. navega para a pasta Não entendi.

Nós da pasta Respostas fazem 3 coisas: redefinem a contagem de vezes consecutivas que o robô não entendeu a entrada para 0 (a contagem é utilizada a seguir); retornam a resposta; e verificam se a próxima mensagem do usuário ativa a intenção “fonte”, que é quando o usuário pede a fonte de uma informação. Em caso positivo, retornam a(s) fonte(s) no formato de links da internet ou respondem que não há uma fonte específica para aquela resposta.

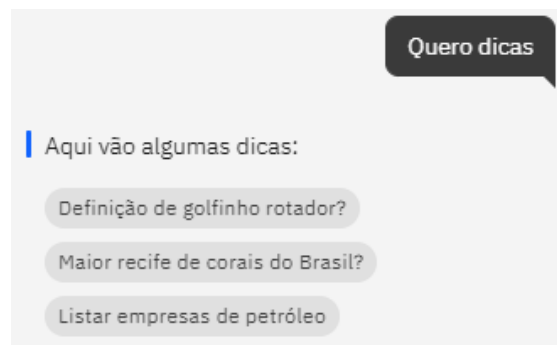
A pasta Não entendi teve seu comportamento descrito apenas em forma de texto, não diagrama, devido à simplicidade. Ela possui o seguinte fluxo (que também se encerra assim que responde algo):

1. verifica se a quantidade de palavras na pergunta (separadas por espaço) é menor que 3, caso sim, afirma que não entendeu e pede ao usuário para digitar perguntas completas;
2. aumenta a contagem de vezes consecutivas que não entendeu;

3. verifica se a contagem é 1, caso sim, afirma que não entendeu e pede ao usuário que reformule;
4. verifica se a contagem é 2, caso sim, realiza o passo 4.a;
 - a. afirma que ainda não entendeu e pergunta ao usuário se ele quer dicas do que perguntar, caso sim, redireciona para o nó de ajuda;
5. verifica se a contagem é superior a 2, caso sim, informa o usuário que não tem a resposta desejada e redireciona para o nó de ajuda sem pedir confirmação.

Na pasta de Diálogo, há um nó especial chamado Ajuda, que fornece ao usuário 3 sugestões de pergunta selecionadas de forma aleatória cada vez que o nó é ativado, conforme o exemplo na Figura 5.

Figura 5 – Exemplo de uso da funcionalidade de Ajuda



Fonte: Autor

De forma resumida, o algoritmo desenvolvido permite detectar a intenção de perguntas sem contexto baseado no contexto da última pergunta feita e garante que a detecção automática pelo Watson seja um último recurso, já que ela depende de probabilidades e é mais sujeita a erros que a detecção determinística (através das partes da frase).

6.2.4 Implementação em código

Com a arquitetura definida, implementou-se um código em Python para criar o chatbot de acordo com a API do Watson a partir da planilha de perguntas e respostas. As principais etapas do código são:

1. extrair o conteúdo da versão anterior do chatbot;
2. extrair o conteúdo da planilha;
3. gerar intenções; *
4. gerar entidades; *
5. gerar nós de diálogo; *

6. organizar os nós; *

7. formar o arquivo final, juntando os resultados das etapas anteriores.

Cada etapa marcada com um asterisco (*) será coberta em detalhes nas seções a seguir. As outras são relativamente simples e não requerem mais explicação que o próprio código. Nas etapas 3-5, informações da planilha e do arquivo da versão anterior do chatbot são combinadas. Por exemplo, na etapa 3, a planilha contém intenções apenas de perguntas que não são da pasta de Diálogo, que é programada pela interface. Logo, para contemplar essas também, é necessário concatenar as informações das duas fontes.

O código utiliza as seguintes bibliotecas:

- copy – permite fazer cópias recursivas de objetos mutáveis;
- json – permite ler e escrever arquivos do tipo JSON;
- os, pathlib – fornecem ferramentas para manipular caminhos de arquivos (paths);
- uuid – permite gerar sequências únicas e aleatórias de caracteres;
- re – permite realizar buscas e substituições com expressões regulares;
- pandas – traz a funcionalidade de DataFrames e outras ferramentas poderosas para manipulação de dados;
- dataclasses – traz a funcionalidade de Dataclasses, que são classes para armazenar informações;
- datetime – permite obter e formatar horários;
- ast – permite interpretar código armazenado dentro de *strings*;
- typing – permite adicionar tipagem estática ao código;
- uuid – permite gerar *strings* únicas e aleatórias.

Observação: muito do código, assim como os comentários presentes nele, está escrito em inglês, por dois motivos. Primeiro, para facilitar a leitura e utilização por um público internacional, caso o trabalho seja divulgado. Segundo, por ser mais conveniente, já que muitas das bibliotecas utilizadas, assim como a própria sintaxe da língua, são em inglês.

6.2.4.1 Geração de intenções

Aqui, obtém-se uma lista de dicionários, onde cada um representa uma intenção. Um exemplo está na Figura 6.

Figura 6 – Exemplo de intenção no arquivo JSON

```
{
  "intent": "amazônia-azul--detalhar-importância",
  "examples": [
    {
      "text": "Qual é a importância da Amazônia Azul?"
    }
  ]
},
```

Fonte: Autor

6.2.4.2 Geração de entidades

Aqui, obtém-se uma lista de dicionários, uma para cada coluna: rótulos, Modificador, Substantivo e Recipiente. Em cada um desses dicionários, o valor da chave “values” é uma lista de dicionários, um para cada entrada única naquela coluna. Por exemplo, para a entidade Modificador, um dos “values” produzidos é o da Figura 7.

Figura 7 – Exemplo de valor da entidade Modificador

```
{
  "type": "synonyms",
  "value": "diferença",
  "synonyms": []
},
```

Fonte: Autor

6.2.4.3 Geração de nós de diálogo

Essa é a etapa que implementa especificamente o algoritmo descrito na Seção 6.2.3. Definiu-se uma classe chamada “Node” para abrigar a representação de um nó de diálogo com seus possíveis atributos, como é possível ver na Figura 8.

Figura 8 – Definição da classe “Node”

```
@dataclass
class Node:
    type: str = "standard"
    title: str = None
    conditions: str = None
    context: Dict = None
    output: Dict = None
    dialog_node: str = field(
        default_factory=lambda: f"node_{uuid.uuid4().hex[:16]}", init=False
    )
    parent: str = None
    previous_sibling: str = None
    next_step: Dict[str, str] = None
    fonte: str = None
    intent: str = None
    modificador: str = None
    substantivo: str = None
    recipiente: str = None
    rotulos: str = None
    children: List["Node"] = field(default_factory=list, init=False)
```

Fonte: Autor

A maior parte dos atributos aparece no arquivo final, conforme o exemplo na Figura 9. Os outros (“fonte” e todos os abaixo) são utilizados no código e removidos antes de produzir o arquivo final. Além dos atributos, a classe possui métodos auxiliares, sendo deles o mais importante o *to_list*, que converte o objeto em uma lista de dicionários, um para cada nó (que são o próprio nó e seus filhos, que ficam armazenados no atributo “children”).

Figura 9 – Exemplo de nó de diálogo

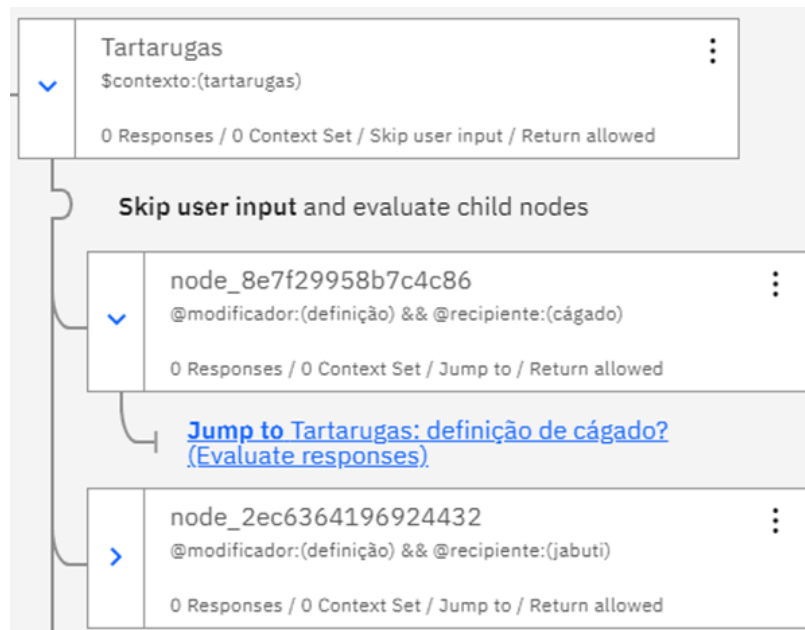
```
{
  "type": "standard",
  "conditions": "@@modificador:(definição) && @recipiente:(actinopterigeo)",
  "dialog_node": "node_26072ef18af64431",
  "next_step": {
    "behavior": "jump_to",
    "selector": "body",
    "dialog_node": "node_b5d0b3dc293a4b83"
  },
  "parent": "node_c126d2e91d764753",
  "previous_sibling": "node_efc71af018b0446f"
},
```

Fonte: Autor

Utilizando a classe Node, o programa gera todos os nós necessários segundo o algoritmo definido anteriormente, já definindo as relações hierárquicas entre eles: “parent”,

“next_step” e “previous_sibling”. Um exemplo dessas relações está na Figura 10. Nela, podemos ver 3 nós de diálogo. O nó do meio possui como “parent” o nó “Tartarugas” e como “next_step” o nó “Tartarugas: definição de cágado?”. Por ser o primeiro filho, ele não possui “previous_sibling”. Já o nó de baixo da imagem possui o mesmo “parent” e seu “previous_sibling” é o do meio.

Figura 10 – Exemplo de hierarquia de nós de diálogo



Fonte: Autor

6.2.4.4 Organização dos nós de diálogo

O objetivo dessa etapa é, principalmente, consertar referências quebradas em nós, por exemplo, campos “parent” que indicam nós que não existem mais. Isso é realizado através de uma classe chamada “NodeOrganizer”. Ela possui como atributo principal um DataFrame e uma série de métodos para manipulá-lo. DataFrames são uma representação bidimensional de dados tabulares (linhas e colunas) e permitem realizar diversas operações em linhas ou colunas inteiras. Nesse caso, eles são usados para representar nós de diálogo, um por linha, sendo suas colunas os atributos da classe Node.

O NodeOrganizer possui 6 métodos principais, que são executados em ordem pela função *run*, como pode ser visto na Figura 11. O primeiro ordena os nós para ficar na mesma ordem que aparecem na interface (ou seja, com base na hierarquia). O segundo remove nós caso seja imposto um limite de número de intenções (o limite varia com o plano contratado do serviço do Watson Assistant). O terceiro altera o nó inicial do diálogo para definir valores

iniciais para uma variável de ambiente, o que é necessário para o funcionamento do nó de ajuda. O quarto altera o nó de ajuda para se adaptar à quantidade de perguntas presentes. O quinto remove e conserta referências quebradas, como nós que apontam para outros que não existem mais nos campos “parent” e “previous_sibling”. O sexto realiza uma referência final que não poderia ter sido feita antes.

Figura 11 – Método *run* da classe *NodeOrganizer*

```
def run(self, intent_limit: int = 0):  
    self.sort_nodes()  
    self.limit_intents(intent_limit)  
    self.set_contexts_node()  
    self.set_help_node()  
    self.fix_previous_siblings()  
    self.point_to_anything_else_node()
```

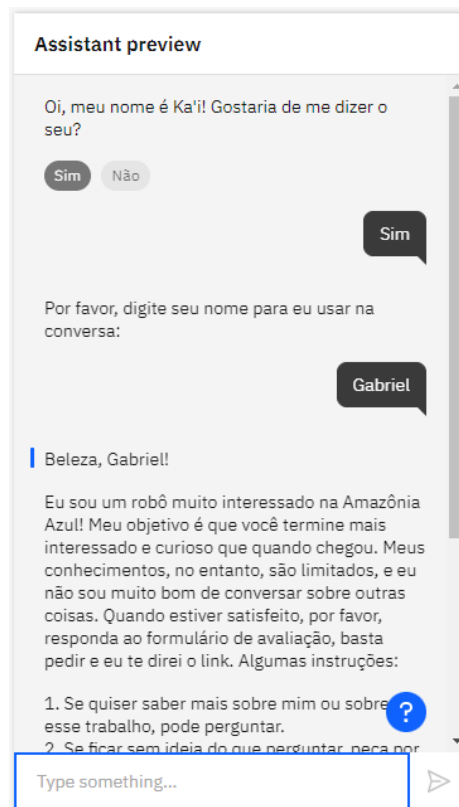
Fonte: Autor

6.3 Distribuição do chatbot

O chatbot completo contém 180 perguntas distribuídas entre 14 temas relevantes à Amazônia Azul.

Depois de concluída a configuração do chatbot, testou-se a integração com o Messenger, aplicativo de comunicação do Facebook. O resultado, porém, foi insatisfatório, devido a algumas limitações de funcionalidade e questões estéticas, que podem afetar a experiência do usuário. Por fim, utilizou-se a ferramenta de *Preview* (visualização) do Watson, que pode ser usado em navegadores tanto de computador quanto de celular, de modo que a acessibilidade foi garantida. A Figura 12 mostra como a tela fica em um navegador de computador.

Figura 12 – Exemplo de conversa através do link de *Preview* do chatbot



Fonte: Autor

RESULTADOS

7 Resultados

Entre os dias 17/11/2021 e 29/11/2021, foram coletadas 46 respostas no formulário de avaliação do chatbot. O chatbot foi compartilhado com um público diversificado em termos de idade, gênero e renda, porém, não há como determinar quem de fato respondeu, pois as respostas foram anônimas.

A Tabela 1 mostra a análise estatística realizada nos dados obtidos. Cada coluna equivale a uma pergunta detalhada na Seção 6.1.2. Os nomes aparecem encurtados, mas as perguntas foram apresentadas completas no formulário. Primeiro, obteve-se a média, o desvio padrão e o tamanho da amostra para cada critério. Em seguida, através do teste t para média de uma amostra, foi possível determinar a menor média estatisticamente provável para cada critério para dois níveis de confiança, 95% e 99%, que são os mais encontrados na literatura.

Tabela 1 – Análise estatística dos resultados do formulário de avaliação

	Entendeu a pergunta	Aprendeu com a experiência	Respostas confiáveis	Melhor que vídeo ou texto	Recomenda a amigos	Satisfação	Expectativas	Média geral
Média do critério	3,28	3,87	4,39	3,04	3,59	3,74	4,02	3,71
Desvio padrão	1,11	1,26	0,93	1,28	1,24	1,12	0,94	1,20
Média esperada	2,5	2,5	2,5	2,5	2,5	2,5	2,5	2,5
Tamanho da amostra	46	46	46	46	46	46	45	45,86
Média mínima (95%)	2,95	3,49	4,11	2,66	3,21	3,4	3,73	3,34
Média mínima (99%)	2,84	3,37	4,02	2,53	3,09	3,29	3,64	3,22
Valor-t a 5%	2,034	2,046	2,051	2,029	2,062	2,046	2,083	2,060
t crítico com 5%	2,015	2,015	2,015	2,015	2,015	2,015	2,016	2,015
Valor-t a 1%	2,707	2,693	2,707	2,717	2,719	2,710	2,724	2,737
t crítico com 1%	2,691	2,691	2,691	2,691	2,691	2,691	2,693	2,691

Fonte: Autor

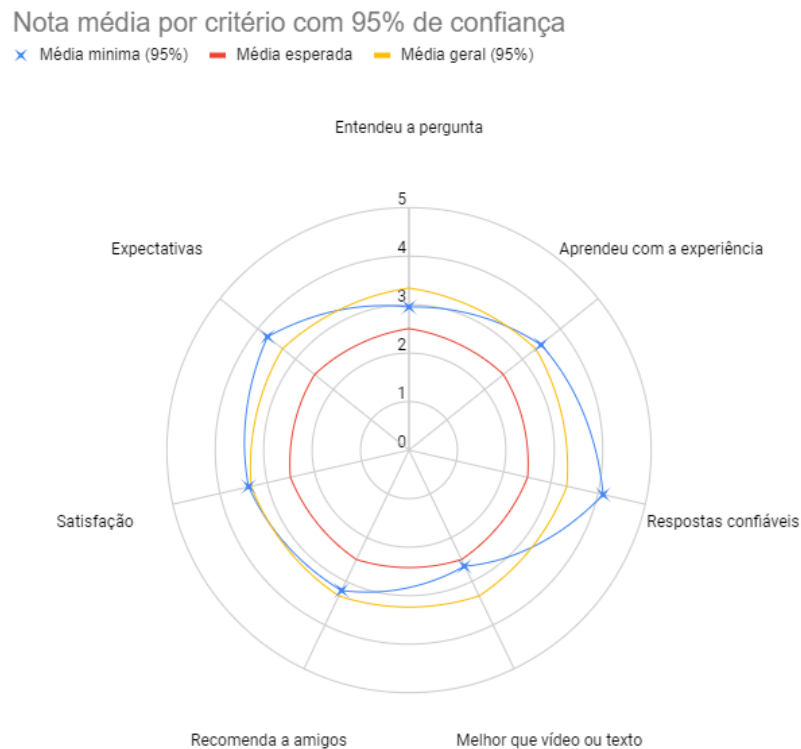
Da maior para a pior média obtida (95%), as perguntas ficam na ordem:

1. Quanto você considera as respostas confiáveis? (4,11)
2. Quanto a experiência atendeu às suas expectativas? (3,73)
3. Quanto você considera que aprendeu com a experiência? (3,49)
4. Qual é seu nível de satisfação com a experiência? (3,40)
5. Quanto você recomendaria esse chatbot a amigos? (3,21)
6. Com que frequência o robô entendeu a sua pergunta? (2,95)
7. Quanto você considerou essa experiência melhor que assistir um vídeo ou ler um texto? (2,66)

Os resultados podem ser visualizados também nos gráficos da Figura 13 e da Figura 14. Mesmo com 99% de confiança, todos os critérios atingiram o objetivo de superar a média de 2,5.

É interessante que a nota de confiabilidade das respostas seja a mais alta, o que demonstra a predisposição de pessoas a confiar no robô, mesmo que não gostem da experiência. Similarmente, notou-se que até aqueles que deram notas ruins no geral ainda consideraram que a experiência atendeu às suas expectativas, ou seja, quem tinha altas expectativas gostou, e quem tinha baixas expectativas não. Esse resultado era esperado conforme a literatura (Seção 4.2).

Figura 13 – Gráfico de radar das médias das respostas do formulário com 95% de confiança



Fonte: Autor

Figura 14 – Gráfico de radar das médias das respostas do formulário com 99% de confiança



Fonte: Autor

Além das 7 perguntas, o formulário tinha um campo aberto de sugestões, onde 23 usuários escreveram algo. Dentre eles:

- 3 (13%) gostariam que os textos fossem acompanhados de fotos
- 3 (13%) sugeriram a ampliação do conjunto de dados (por inteiro ou de um tema específico)
- 2 (8,7%) gostariam da possibilidade de fazer perguntas por voz, ao invés de digitar
- 6 (26%) sugeriram funcionalidades, como guiar mais a experiência, perguntando se o usuário gostaria de saber mais sobre um assunto, ou levar em consideração mais mensagens anteriores
- 4 (17%) apontaram comportamento inesperado, por exemplo, uma pergunta deixou de ser identificada mudando apenas 1 letra

Algumas métricas foram extraídas da interface do Watson, dentre elas:

- Foram realizadas 155 conversas
- Na média, enviou-se 9,48 mensagens por conversa
- No total, enviou-se 1469 mensagens ao chatbot
- A cobertura foi de 44,5%

- No total, 90 pessoas utilizaram o chatbot
- Cada usuário realizou, na média, 1,72 conversas

Quanto aos requisitos do projeto, observou-se o seguinte:

1. Tempo de resposta: o tempo observado médio foi de até 2 segundos, ficando bem dentro da margem de 5 segundos.
2. Abranger perguntas sobre a AAz e certos temas: os temas foram abordados na seguinte proporção: poluição (1), correntes oceânicas (1), litoral (1), turismo (2), flora (3), gás (3), oceano (5), baleias (5), energia de maré (7), marés (7), animais marinhos (7), pesca (9), Amazônia Azul (14), corais (19), tartarugas (20), outras (21), petróleo (25) e fauna (34), totalizando 180 perguntas (4 delas em mais de um tema).
3. Providenciar uma experiência positiva aos usuários: em acordo com o definido na Seção 6.1.2, as médias obtidas foram superiores a 2,5.
4. Contribuir para a educação dos usuários: mesmo que o acima.
5. Acompanhar o contexto da conversa: o robô é capaz de entender uma pergunta sem contexto baseado no contexto da pergunta prévia.
6. Dar respostas curtas: as respostas foram limitadas a um comprimento que cabe na tela da maioria dos dispositivos móveis sem precisar rolar.
7. Permitir o usuário guiar a conversa: o usuário recebe uma curta introdução, depois disso é livre para perguntar o que quiser e pode pedir ajuda a qualquer momento.
8. Ser de fácil acesso: a distribuição em um site próprio não apresentou obstáculos para o uso do chatbot.

Logo, todos foram cumpridos.

CONCLUSÃO

8 Conclusão

Esse trabalho construiu uma base de dados introdutórios sobre a Amazônia Azul, reunindo informações confiáveis de diversas fontes e abordando diversos dos seus principais temas. Além disso, ele produziu um chatbot capaz de ensinar, superar expectativas e trazer satisfação. Através de avaliações de usuários, o chatbot demonstrou ser uma ferramenta viável e bem recebida para contribuir para a educação dos brasileiros sobre a Amazônia Azul.

O trabalho também contribui para avançar os conhecimentos atuais sobre chatbots, uma área de estudo que vem se popularizando nos últimos anos. Em especial, existe uma carência por métodos padronizados para avaliação da qualidade e do sucesso desse tipo de projeto. Nesse quesito, o trabalho presente, ao se basear nos estudos mais recentes sobre a situação de métodos de avaliação, ajuda a constituir uma base para trabalhos futuros, que poderão se inspirar nos critérios utilizados e usar os valores obtidos como referência.

Outra contribuição importante é a arquitetura e o código elaborados para a implementação do chatbot. Esse conjunto de técnicas e ferramentas pode ser considerado a base de um *framework* para a construção de chatbots de diálogo, especialmente os de perguntas e respostas, que precisam de soluções escaláveis para lidar com centenas de intenções e ainda mais elocuições.

8.1 Considerações para trabalhos futuros

De acordo com testes realizados e feedback de usuários, seria interessante:

- Permitir que perguntas diferentes apontem para a mesma resposta – isso se refere a perguntas que, apesar de tecnicamente diferentes, possuem apenas uma resposta no banco de dados, por se considerar que suas respostas são semelhantes. Por exemplo, a resposta de “Quantas espécies há na Amazônia azul?” já inclui em seu corpo a divisão dos números entre peixes, crustáceos e moluscos, então se o usuário perguntar “Quantas espécies de peixes há na Amazônia Azul?”, a mesma resposta pode ser utilizada.
- Permitir que perguntas apontem para a combinação de mais de uma resposta – esse recurso poderia ser utilizado no caso de perguntas como “Qual a diferença entre tartarugas e jabutis?”, onde uma nova resposta foi criada, porém seria aceito concatenar as respostas de “O que é uma tartaruga?” e “O que é um jabuti?”, que já destacam características comparáveis entre os animais.

- Popular a base de dados com mais informações – esse foi um pedido comum entre usuários que participaram da avaliação do chatbot.
- Desenvolver um algoritmo (estatístico ou de aprendizado de máquina) que obtenha Contexto, Modificador, Substantivo e Recipiente automaticamente para montar a base de dados – conforme mencionado na Seção 6.2.2, isso permitiria acelerar o processo de classificar perguntas, necessário para a detecção automática de elocuições.
- Quando um contexto for identificado mas uma intenção não, sugerir perguntas daquele contexto – isso melhoraria a experiência dos usuários, visto que alguns apontaram na avaliação que gostariam que ela fosse mais guiada.
- Quando o usuário pedir dicas, dar prioridade aos temas menos explorados – novamente, uma funcionalidade para tornar a experiência mais guiada.
- Oferecer uma função “saber mais” para algumas perguntas, que permite o usuário requisitar mais informações sobre aquela resposta – essa também foi uma funcionalidade requisitada por usuários que poderia melhorar a experiência.

REFERÊNCIAS

- AI, D. (2019). *Machine Learning Glossary and Terms*. <https://deepai.org/machine-learning-glossary-and-terms/conversational-agent> (Acessado em: 12/05/2021)
- Arbaugh, J. B. (2001). How Instructor Immediacy Behaviors Affect Student Satisfaction and Learning in Web-Based Courses. *Business Communication Quarterly*, 64(4), 42–54. <https://doi.org/10.1177/108056990106400405>
- Bansal, S. (2016). *Beginners Guide to Topic Modelling in Python*. <https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/> (Acessado em: 21/07/2021)
- Blei, D., Carin, L., & Dunson, D. (2010). Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6), 55–65. <https://doi.org/10.1109/MSP.2010.938079>
- Brennan, S. E. (1991). Conversation with and through computers. *User Modeling and User-Adapted Interaction*, 1(1), 67–86. <https://doi.org/10.1007/BF00158952>
- Casas, J., Tricot, M. O., Abou Khaled, O., Mugellini, E., & Cudré-Mauroux, P. (2020). Trends & methods in chatbot evaluation. *ICMI 2020 Companion - Companion Publication of the 2020 International Conference on Multimodal Interaction, October*, 280–286. <https://doi.org/10.1145/3395035.3425319>
- Castro, B. M., Brandini, F. P., & Fortes, J. F. (2017). *A Amazônia Azul: recursos e preservação*. 7–26.
- Gnewuch, U., Morana, S., Adam, M. T. P., & Maedche, A. (2018). Faster Is Not Always Better: Understanding the Effect of Dynamic Response Delays in Human-Chatbot Interaction the 26 th European Conference on Information Systems (ECIS), Portsmouth , United Kingdom , June 23-28 . Please note : Copyright is owned by th. *Proceedings of the 26th European Conference in Information Systems (ECIS), June*.
- Grootendorst, M. (2020). *Topic Modelling with BERT*. <https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6>
- IBM. ([s.d.]). *Getting started with Watson Assistant*. <https://cloud.ibm.com/docs/assistant> (Acessado em: 12/05/2021)
- IBM. (2020). *What is Application Programming Interface (API)*. <https://www.ibm.com/cloud/learn/api> (Acessado em: 01/12/2021)
- International Standards Organization. (2018). *ISO 9241*. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>

- Jain, M., Kumar, P., Kota, R., & Patel, S. N. (2018). Evaluating and informing the design of chatbots. *DIS 2018 - Proceedings of the 2018 Designing Interactive Systems Conference*, 895–906. <https://doi.org/10.1145/3196709.3196735>
- Kerly, A., Hall, P., & Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, 20(2), 177–185. <https://doi.org/10.1016/j.knosys.2006.11.014>
- Koehrsen, W. (2018). *Neural Network Embeddings Explained*. <https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526> (Acessado em: 20/07/2021)
- Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22 140, 55.
- Market Research Institute International. ([s.d.]). *Questionnaire Design*. https://www.insightsassociation.org/sites/default/files/misc_files/mrii_questionnaire_design_webinar_for_ia_v4.pdf (Acessado em: 01/12/2021)
- Pereira, R. (2019). *O que é a Amazônia Azul e por que o Brasil quer se tornar potência militar no Atlântico*. <https://www.marinha.mil.br/economia-azul/noticias/o-que-é-amazônia-azul-e-por-que-o-brasil-quer-se-tornar-potência-militar-no-atlântico> (Acessado em: 09/05/2021)
- Qi, H., Pan, L., Sood, A., Shah, A., Kunc, L., & Potdar, S. (2020). Benchmarking intent detection for task-oriented dialog systems. *arXiv*, 150.
- Radziwill, N. M., & Benton, M. C. (2017). *Evaluating Quality of Chatbots and Intelligent Conversational Agents*. April. <http://arxiv.org/abs/1704.04579>
- Rajaraman, A., & Ullman, J. (2011). Data Mining. In *Mining of Massive Datasets* (p. 1–17). <https://doi.org/https://doi.org/10.1017/CBO9781139058452.002>
- Reimers, N., & Gurevych, I. (2020). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. <https://doi.org/10.18653/v1/d19-1410>
- Riehle, D. (2000). Framework Design: A Role Modeling Approach. *Softwaretechnik-Trends*, 20.
- Saris, W. E., Revilla, M., Krosnick, J. A., & Shaeffer, E. M. (2010). Comparing questions with agree/disagree response options to questions with item-specific response options. *Survey Research Methods*, 4(1), 45–59. <https://doi.org/10.18148/srm/2010.v4i1.2682>

- Söllner, M., Bitzer, P., Janson, A., & Leimeister, J. M. (2018). Process is king: Evaluating the performance of technology-mediated learning in vocational software training. *Journal of Information Technology*, 33(3), 233–253. <https://doi.org/10.1057/s41265-017-0046-6>
- Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>
- Williams, J., Raux, A., Ramachandran, D., & Black, A. (2013). The dialog state tracking challenge. *SIGDIAL 2013 - 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Proceedings of the Conference*, 404–413.
- Winkler, R., & Söllner, M. (2018). Unleashing the Potential of Chatbots in Education : A State-Of-The-Art Analysis. *Academy of Management Annual Meeting Proceedings*. <https://doi.org/http://dx.doi.org/10.5465/AMBPP.2018.15903abstract>
- Yin, J., Goh, T. T., Yang, B., & Xiaobin, Y. (2021). Conversation Technology With Micro-Learning: The Impact of Chatbot-Based Learning on Students' Learning Motivation and Performance. *Journal of Educational Computing Research*, 59(1), 154–177. <https://doi.org/10.1177/0735633120952067>