Desenvolvimento de um Modelo *Ensemble* para Responder Questões via Bancos de Dados Estruturados (KB-QA)

Rafael H. de Sousa [*] **Bruno H. G. Barbosa** [**] **Danton D. Ferreira** [**]
Paulo R. Silva [***] **Sinval T. Nascimento** [***]

[*] *Programa de Pós-Graduação em Engenharia de Sistemas e Automação, Universidade Federal de Lavras (e-mail: rafael.henrico10@gmail.com)*
[**] *Departamento de Automática, Universidade Federal de Lavras, CP 3037, 37200-900, Lavras/MG (e-mails: brunohb@ufla.br, danton@ufla.br).*
[***] *Omnilogic Inteligência S/A, Belo Horizonte/MG (e-mails: paulo.silva@omnilogic.ai, sinval@omnilogic.ai).*

Abstract: The virtual sales market (E-commerce) has expanded a lot nowadays due to the ease and practicality provided by this way of purchase, and the fact that technologies are becoming increasingly accessible. With this, the implementation of virtual assistants by companies can add benefits for both sides of the negotiation (consumers-companies), since the use of virtual assistants can allow the automation of tasks (involving means of communication), thus accelerating solving problems and increasing productivity for customer service, in addition to being able to provide personalized experiences tailored to each customer. In this work we discuss some models and strategies that are relevant to the topic of Knowledge Base Question Answering (KB-QA), and means of developing a KB-QA model capable of answering user questions provided in natural language, based on information contained in knowledge bases (KBs) that can be implemented as part of a virtual assistant. In order to develop a superior KB-QA model via Ensemble by majority vote, the search, analysis and selection of KB-QA models are carried out for the Ensemble's composition. For the analysis of the KB-QA models, the WebQuestionsSP question bank is used, which is developed to be answered through queries to the Freebase KB. As the main result of this work, three different Ensembles are generated, called Simple Ensemble, Ensemble with Countermeasure and Ensemble with Total Countermeasure, which are state-of-the-art for the KB-QA task considering the WebQuestionsSP database with F1-scores of 75.40%, 78.72% and 81.43% respectively.

Resumo: O mercado de vendas virtuais (*E-commerce*) tem se expandido muito atualmente devido à facilidade e praticidade proporcionadas por esta via de compra e pelo fato de que as tecnologias vem se tornando cada vez mais acessíveis. Com isso, a implantação de assistentes virtuais por empresas pode agregar benefícios para ambos os lados da negociação (consumidores-empresas), uma vez que o uso de assistentes virtuais pode permitir a automação de tarefas (envolvendo meios de comunicação), acelerando assim a resolução de problemas e aumentando a produtividade para atendimento ao cliente, além de ser capaz de fornecer experiências personalizadas adequadas a cada consumidor. Neste trabalho, são abordados alguns modelos e estratégias que são relevantes ao tópico de *Knowledge Base Question Answering (KB-QA)*, e meios de desenvolver-se um modelo *KB-QA* capaz de responder a questões do usuário fornecidas em linguagem natural, baseado em informações contidas em bases de conhecimento (*KBs*) que possa ser implementado como parte de um assistente virtual. A fim de se desenvolver um modelo *KB-QA* superior via *Ensemble* por voto majoritário, realiza-se a busca, análise e seleção de modelos *KB-QA* para a sua composição. Para análise dos modelos *KB-QA* emprega-se o banco de questões *WebQuestionsSP* que é desenvolvido para ser respondido através de consultas à *KB Freebase*. Como principal resultado deste trabalho, são gerados três diferentes *Ensembles*, denominados *Ensemble* simples, *Ensemble* com contramedida e *Ensemble* com contramedida total, que são estado-da-arte para a tarefa de *KB-QA* considerando-se o banco de dados *WebQuestionsSP* com F1-scores de 75,40%, 78,72% e 81,43% respectivamente.

Keywords: Knowledge Base Question Answering; KB-QA; Virtual Assistants; Natural Language Processing; Artificial Intelligence; Chatbots; E-commerce.

Palavras-chaves: Knowledge Base Question Answering; KB-QA; Assistentes Virtuais; Processamento de Linguagem Natural; Inteligencia Artificial; Chatbots; E-commerce.

1. INTRODUÇÃO

Nos últimos anos, o mercado brasileiro de vendas virtuais tem se expandido cada vez mais. No ano de 2020, acelerado devido às restrições impostas para o varejo físico em função da pandemia de Covid-19 e a necessidade de distanciamento social, o comércio eletrônico brasileiro apresentou um crescimento de 68% em relação ao ano anterior, no ano de 2021 ocorreu um crescimento de 19% em relação a 2020, com uma projeção de crescimento de 12% para o ano de 2022 (ABCOMM, 2021). Com isso, a implantação de assistentes virtuais ou de *chatbots* se torna uma opção viável para a redução dos custos empresariais, pois permitem a resolução de problemas, o aumento da produtividade no atendimento ao cliente e entrega de experiências personalizadas para cada cliente, graças às informações fornecidas pelos próprios consumidores às empresas.

O desenvolvimento de um assistente virtual é composto de várias etapas, que consistem desde a compreensão da mensagem em linguagem natural emitida pelo usuário, até o fornecimento de respostas ou opiniões de acordo com a necessidade do usuário, em conformidade com o diálogo estabelecido. Após a interpretação da mensagem do usuário, caso seja uma solicitação de informação, o assistente virtual precisa determinar a resposta à questão que deverá ser fornecida ao usuário (Zhou et al., 2020).

Como fonte de conhecimento, os assistentes virtuais podem buscar as repostas em bancos de dados de textos não estruturados ou bancos de dados estruturados denominados *Knowledge Bases (KB)*, cuja informação pode ser vista como uma estrutura de grafo, também conhecida como *Knowledge Graph*. As *KBs* permitem a inferência de informações em grafos e/ou tabelas, e reduzem a demanda de armazenamento pelo banco de dados (Gao et al., 2018). Considerando-se estes benefícios das consultas em *KBs*, enfatiza-se esta tarefa ao longo deste trabalho.

A tarefa de responder a questões com base em informações contidas em *Knowledge Bases* é denominada *Knowledge Base Question Answering (KB-QA)*. O desenvolvimento de um modelo *KB-QA* capaz de responder a uma maior variedade de questões corretamente é importante para o comércio eletrônico, uma vez que um assistente virtual que contenha um bom modelo *KB-QA* será capaz de suprir uma maior quantidade de demandas de informação e suporte provenientes dos usuários/clientes, o que garantiria para a empresa, por exemplo, uma menor necessidade de pessoal dedicado a atendimento.

Para desenvolvimento de modelos voltados para a consulta de informações em *Knowledge Bases (KB-QA)*, diversas abordagens são listadas na literatura, principalmente relacionadas a forma a qual a necessidade de informação contida na questão é representada pelo sistema para consulta à *KB* e como é realizada a busca pelo sistema. As principais abordagens são a abordagem de análise semântica, a abordagem de recuperação de informação e a abordagem de análise semântica neural (Fu et al., 2020).

A abordagem de análise semântica aplica regras ou *templates* desenvolvidos com base em informações semânticas e/ou sintáticas das questões para formação de uma representação da forma lógica da questão (em geral um grafo de representação da questão). Um exemplo de modelo nesta linha é o *GAnswer* (Hu et al., 2017) que apresenta duas estratégias para a formação da forma lógica, *node first* que prioriza a detecção das entidades da questão na sua representação, e a estratégia *relation first* que prioriza a detecção das relações.

Outro modelo que emprega este tipo de abordagem é o *Parot* (Ochieng, 2020). O *Parot* desenvolve várias regras baseadas nas árvores de dependências das questões para obtenção de uma representação em forma de *Sparql Protocol And Rdf Query Language* (*SPARQL*), uma linguagem padronizada que pode ser usada para consultar à *KB*s no formato de grafos *RDF (Resource Description Framework)*. Um grande problema com a abordagem de análise semântica é que gerar regras ou *templates* é trabalhoso, e representar alguns tipos de questões pode ser inviável.

A estratégia de recuperação de informação consiste de representar as questões e informações da *Knowledge Base* através de redes neurais, bem como também são criadas as representações das informações relacionadas a *Knowledge Base*. Para isto, são empregadas complexas estruturas de redes. Um modelo interessante que aplica este tipo de estratégia é o *Bidirectional Attentive Memory Networks (BAMnet)* (Chen et al., 2019), que representa a questão através de *embeddings* e então utiliza redes como *Long Short Term Memory (LSTM)* e mecanismos de atenção para refinamento das representações em conjunto com representações da *KB* via *Key-Value Memories*.

Na mesma linha de pesquisa, o *Neural State Machine teacher-student (NSM)* (He et al., 2021) aplica máquinas de estado neural para fazer a representação da questão e da *Kowledge Base* através da técnica *teacher-student* onde a rede *teacher* é treinada para conseguir fornecer informações intermediárias para direcionar a rede *student* na busca da resposta para a questão, uma vez que em geral os modelos só possuem informação da resposta final para treino, sem informações sobre passos intermediários nos grafos. Modelos de recuperação de informação sofrem com erros devido a modelagem de restrições, devido a suas formas de seleção de respostas (geralmente baseada em limiares) dentre outros.

Já a estratégia de análise semântica neural visa combinar os modelos anteriores por meio de métodos neurais para gerar representações em forma lógica das questões. Atualmente tais modelos tem apresentado os melhores resultados para a tarefa de *KB-QA* e estão dentre o modelos estado-da-arte para os principais bancos de dados

da literatura. Dois modelos interessantes que empregam esta estratégia são o *Neural-Symbolic Complex Question Answering (NS-CQA)* (Hua et al., 2020) e o *Multi-hop Complex KBQA (Multihop)* (Lan and Jiang, 2020), ambos empregam técnicas de aprendizado por reforço para gerar as representações das questões, se diferindo principalmente na forma como suas ações de geração do grafo são desenvolvidas.

Apesar das diferentes abordagens e sua evolução, a tarefa *KB-QA* ainda demonstra uma brecha para grande aprimoramento. Atualmente na literatura, tem-se focado em criar modelos que possam lidar com todas as questões simultaneamente independentemente da estratégia empregada. É importante notar, no entanto, que existe um grande desafio em lidar com questões em linguagem natural (que pode conter erros, faltar informação de suporte ou possuir altas complexidades), além de que existem muitos processos diferentes para interpretação destas informações pela máquina, o que torna um problema de grande espectro de possibilidades. Além disto, há dificuldades relacionadas à inferência da resposta na *KB* que pode faltar informações ou apresentar padrões irregulares de representação das informações.

Para tentar contornar tais dificuldades e gerar um modelo *KB-QA* que possa resultar em um assistente virtual mais robusto para fornecer respostas, neste trabalho desenvolve-se um modelo *KB-QA* via *Ensemble* por voto majoritário empregando-se três diferentes modelos individuais para *KB-QA*, através de três abordagens denominadas aqui por simples, com contramedida e com contramedida total, que visam suprir as limitações dos modelos individuais através de suas diversidades e coesões, a fim de obter um melhor desempenho em termos de *F1-score*. O texto a seguir está organizado da seguinte maneira: na Seção 2 são apresentadas as ferramentas, a descrição do *Ensemble* e abordagens empregadas, na Seção 3 são mostrados os resultados e suas análises, e por fim, na Seção 4 são apresentadas as conclusões do trabalho e sugestões para trabalhos futuros.

2. MATERIAIS E MÉTODOS

2.1 Banco de Dados

O banco de dados de questões tomado como referência e empregado para avaliação dos modelos individuais e do *Ensemble* é o banco *WebQuestionsSP*, que é uma variação do banco de dados de questões *WebQuestions* de onde foram removidas algumas questões que não poderiam ser respondidas por meio de *SPARQLs* (Yih et al., 2015). Além disso, são atribuídos para cada questão um padrão *SPARQL* que pode ser usado como referência em modelos que utilizam-se de *SPARQLs* para consulta à *Knowledge Base* além de outras informações de análise semântica das questões (*Semantic Parsing (SP)*).

O *WebQuestionsSP* contém uma coleção de questões *multi-hop* (questões que necessitam de uma cadeia de triplas da *KB* para serem respondidas), sobre assuntos gerais, cujas respostas podem ser uma entidade ou um conjunto de entidades. Um análise das complexidades das questões é mostrada na Tabela 1. O banco possui 4737 pares de questões-respostas que são divididos em um conjunto de treino com 3097 questões e um conjunto de teste com 1639 questões (Hua et al., 2020).

Além disto, neste trabalho foram selecionadas um total de 550 questões do conjunto de treino para compor o conjunto de validação a ser usado posteriormente no modelo de *Ensemble* simples, com a finalidade de computar seus F1-scores neste conjunto de validação para uso como fator ponderador nas construções dos modelos de *Ensembles* com contramedidas. As questões colocadas no conjunto de validação são as mesmas empregadas como conjunto de validação no trabalho de Chen et al. (2019), removendo-se as questões que não pertencem ao banco *WebQuestionsSP*.

Tabela 1. Estatísticas de complexidade das questões contidas no *WebQuestionsSP*

Tipo de Questão	Percentual das Questões
1-hop s/ restrição	71,3%
1-hop c/ restrição	28,2%
2-hop s/ restrição	0,0%
2-hop c/ restrição	0,5%

Fonte: Adaptado de Lan and Jiang (2020)

As questões contidas no *WebQuestionsSP* são desenvolvidas para que possam ser respondidas com base em informações contidas na *Knowledge Base* denominada *Freebase* criada pelo *Google* e que contém cerca de 2,6 bilhões de triplas sobre aproximadamente 44 milhões de tópicos (Yih et al., 2015). Devido ao tamanho desta *KB*, em geral, o que é feito pelos modelos é uma extração de subgrafos que estejam em uma determinada vizinhança das entidades tópicos determinadas pelos modelos para as questões e então realiza-se a consulta nestes subgrafos.

2.2 Ferramentas KB-QA

Nesta Sub-seção é realizada uma breve descrição do funcionamento dos modelos *KB-QA* individuais que compõem o *Ensemble* por voto majoritário desenvolvido, nomeados aqui *BAMnet* (Chen et al., 2019), *NS-CQA* (Hua et al., 2020) e *Multihop* (Lan and Jiang, 2020). Os modelos foram selecionados após uma revisão bibliográfica e reavaliação de seus resultados por meio de suas implementações, seguindo as instruções ótimas de seus artigos. Para uma visão mais detalhada e acesso as equações, vide os artigos originais dos modelos.

BAMnet. O *BAMnet* (*Bidirectional Attentive Memory Network*) desenvolvido por Chen et al. (2019) é um modelo *KB-QA* baseado em *embeddings*, ou seja, todas as operações desde a conversão da questão em linguagem natural, seleção das características e busca da resposta na *Knowledge Base* são realizadas no espaço neural. O *BAMnet* pode ser descrito em geral como quatro macro-módulos, sendo eles, o *input module*, o *memory module*, o *reasoning module* e o *answer module*:

(1) *Input module*: módulo de entrada do modelo. Uma questão é representada como uma sequência de *embeddings* das palavras aplicando-se uma *embedding layer*. Então emprega-se *Bidirectional Long Short Term Memory (BiLSTM)* para gerar uma representação da questão;
(2) *Memory module*: Neste módulo, seleciona-se os candidatos a resposta (etapa *candidate generation)* e

então gera-se uma representação da *Knowledge Base* para cada candidato através de uma rede *Key-Value Memory*;
(3) *Reasoning module*: é o módulo responsável por interpretar os dados, a questão e suas respostas candidatas, e refinar a seleção da resposta, constituído pelo módulo de generalização, e a rede bidirecional de duas camadas.
(4) *Answer module*: módulo para seleção da resposta. Baseado na representação da questão e a representação das respostas candidatas, utiliza-se uma medida de similaridade para ranquear os candidatos e então selecionar as respostas.

NS-CQA. O modelo *NS-CQA (Neural-Symbolic Complex Question Answering)*, desenvolvido por Hua et al. (2020), emprega o método de análise semântica neural. De forma simplificada, o *NS-CQA* consiste de dois módulos básicos para lidar com as questões e respondê-las, o Gerador Neural, com a função de transformar a questão em linguagem natural em uma sequência de ações primitivas e o Executor Simbólico, com a função de executar as ações primitivas geradas para a questão contra a *Knowledge Base* a fim de se determinar a resposta da questão. A estrutura do *NS-CQA* pode ser dividida em três grandes estágios de tarefas, onde o primeiro estágio são as tarefas que antecedem o módulo do Gerador Neural. Estes estágios são descritos de forma resumida a seguir:

(1) Analisador Semântico: este estágio é responsável por reconhecer artefatos da *Knowledge Base* que são relevantes para a questão. Dada uma questão em linguagem natural, primeiramente são reconhecidas menções a entidades e a classes de entidades. Feito isto, as entidades e tipos são referenciadas com as correspondentes entidades e tipos contidas na *KB* utilizando-se similaridade literal e similaridade semântica. Após isto, as entidades e tipos na questão são substituídos por coringas para geração de padrões. São gerados ainda padrões para as correspondentes relações contidas na questão.
(2) Gerador Neural: este estágio é responsável por gerar o conjunto de ações primitivas, baseado em um modelo de atenção *Sequence to Sequence (Seq2Seq)* aumentado com mecanismos de cópia e mascaramento, o que permite a redução do tamanho do vocabulário do decodificador e alivia problemas devido a palavras fora do vocabulário. O modelo trabalha com um conjunto de 17 tipos de ações primitivas que podem ser vistas em detalhes no trabalho de Hua et al. (2020).
(3) Executor Simbólico: este é o estágio responsável por executar as ações primitivas geradas pelo Gerador Neural. Primeiramente, o Executor Simbólico analisa os *tokens* de saída produzidos pelo Gerador Neural e monta as ações uma a uma. Após isso, dada a sequência de ações gerada, a partir da primeira ação, o Executor Simbólico as executa em ordem, uma a uma, operando sobre os resultados intermediários da ação anterior até que se encontre a ação final, onde o resultado do último passo de execução é retornado como a resposta definitiva.

Multihop. Aqui é realizada uma breve descrição do modelo desenvolvido no trabalho de Lan and Jiang (2020). O modelo, denominado aqui por *Multi-hop Complex Questions* ou apenas *Multihop*, é uma abordagem criada a fim de lidar com questões complexas em linguagem natural. A abordagem feita por Lan and Jiang (2020) visa lidar com questões com restrições e que necessitam de múltiplos passos de busca simultaneamente, através de uma abordagem de análise semântica combinada com aprendizado por reforço. O funcionamento completo do modelo, desde o processo de geração do *query graph* (representação em forma lógica da questão) até a busca da resposta, pode ser genericamente descrito nas seguintes quatro etapas:

(1) Partindo-se de uma entidade básica encontrada na questão (denominada entidade tópico), identificar um caminho relacional central que conecte a entidade tópico a uma variável *lambda*;
(2) A partir do caminho relacional central encontrado no passo anterior, adicionar uma ou mais restrições encontradas na questão. Uma restrição pode ser uma entidade básica ou uma função de agregação junto com uma relação;
(3) Com todos os *query graphs* candidatos gerados nos dois passos anteriores, ranqueá-los medindo-se suas similaridades com relação à questão;
(4) Executar os *query graphs* melhor ranqueados contra a *KB* para obter as entidades respostas.

2.3 Ensemble (Comitê por Voto Majoritário)

Diversas técnicas podem ser empregadas para combinação e aperfeiçoamento de modelos envolvidos nas áreas de aprendizado de máquina, e dentre essas técnicas, os *Ensembles* são considerados um avanço da área de *machine learning* devido às vantagens oferecidas em diversos tipos de situações, desde que os modelos independentes sejam suficientemente acurados e demonstrem diversidade.

Para desenvolvimento de um modelo *KB-QA* aprimorado é aplicada a técnica de *Ensemble* por voto majoritário. Desta forma após a análise de modelos completos encontrados na literatura, viáveis de reimplementação, foram selecionadas distintas combinações de 3 modelos para composição de diferentes *Ensembles* a fim de se avaliar qual combinação garante o melhor desempenho com relação ao banco de dados *WebQuestionsSP* (Subseção 2.1) de acordo com as métricas descritas na Subseção 2.4.

Uma ilustração da estrutura do *Ensemble* por voto majoritário desenvolvido pode ser vista na Figura 1. Previamente, os modelos são treinados individualmente para que possam responder as questões contidas no banco de dados *WebQuestionsSP*. Desta forma, cada modelo individual pode ser visto como uma das pessoas a responder a questão na Figura 1. O funcionamento do *Ensemble* é dado da seguinte forma:

a) 1º passo: o usuário fornece uma questão ao sistema;
b) 2º passo: a questão é respondida pelos modelos individuais separadamente;
c) 3º passo: são verificados quais das respostas são comuns entre a maioria dos modelos (entre 2 ou mais). Na implementação é necessário verificar a existência de consenso para cada resposta distinta gerada pelos modelos, e como algumas métricas podem ser influenciadas pela ordem das respostas (vide Subseção 2.4),

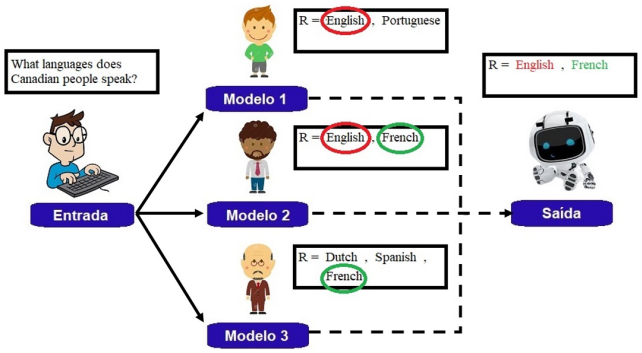

Figura 1. Estrutura do ensemble: Exemplo de funcionamento para uma questão

a ordem de prioridades das votações das respostas é dada em ordem decrescente de acordo com os *F1-scores* dos modelos individuais no conjunto de validação;

d) 4º passo: as respostas com consenso são então selecionadas como o conjunto de respostas final.

O modelo descrito até aqui é denominado neste trabalho como *Ensemble* simples ou simplesmente *Ensemble*. Esta abordagem, no entanto, permite ainda que haja muitas lacunas de questões não respondidas pelo modelo, uma vez que podem haver questões em que não haja consenso de respostas entre os modelos individuais. Para lidar com isto, faz-se duas abordagens para reduzir o número de questões sem respostas pelo modelo:

a) Contramedida simples: nesta abordagem é feita uma reavaliação do *Ensemble* simples para o conjunto de validação extraído do conjunto de treino de *WebQuestionsSP*. Em seguida, os modelos individuais são reavaliados através da métrica *F1-score* para as questões do conjunto de validação que não possuem consenso. Desta forma, o modelo que possui maior *F1-score* para os casos de não consenso é selecionado como o modelo individual de contramedida, ou seja, nos casos de não consenso para uma questão, as respostas do modelo individual de contramedida são fornecidas como a resposta final do *Ensemble*;

b) Contramedida total: no caso de contramedida simples, o modelo selecionado como contramedida para o caso de não consenso das respostas pode também ser incapaz de responder a algumas questões (fornecendo resposta vazia). A abordagem de contramedida total visa minimizar ao máximo a existência de respostas vazias, desta forma, de acordo com a reavaliação dos modelos individuais para as questões sem consenso de respostas no conjunto de validação, é criada uma lista de prioridades dos modelos para responder as questões sem consenso com base neste *F1-score*. Assim, quando não há consenso entre os modelos, primeiramente o modelo individual com maior *F1-score* nos casos de não consenso é selecionado para responder as questões, caso ele não seja capaz, seleciona-se o segundo melhor modelo (com base no F1-score nestes casos) e assim sucessivamente até que não haja opções de modelos que possam responder as questões dentre os modelos que compõem o *Ensemble*.

Os resultados do melhor *Ensemble* com contramedida simples (também denominado como *Ensemble* com contramedida) e do melhor *Ensemble* com contramedida total são apresentados na Seção 3.

2.4 Ferramentas de Análise Estatística

A precisão é a métrica que indica o percentual de respostas que são corretas dentre as respostas retornadas pelo modelo para uma questão. Desta forma, a precisão média de um modelo pode ser definida como,

$$Precisão = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{c_i}{t_i}\right), \tag{1}$$

em que m é o número de questões avaliadas, c_i é o número de respostas corretas retornadas pelo modelo para a questão i e t_i é o número total de respostas retornadas pelo modelo para a questão i.

O *Recall* demonstra o percentual das respostas corretas que o modelo é capaz de recuperar para uma questão. Assim, o *Recall* médio de um modelo é definido como,

$$Recall = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{c_i}{n_i}\right), \tag{2}$$

em que m é o número de questões avaliadas, c_i é o número de respostas corretas retornadas pelo modelo para a questão i e n_i é o número de respostas corretas existentes para a questão i.

A principal métrica de avaliação de desempenho empregada nesta pesquisa é o *F1-score*, que é uma ponderação entre a Precisão e o *Recall* médios (também denominada Macro *F1-score*), definida como,

$$F1\text{-}score = \frac{1}{m}\left(\sum_{i=1}^{m}\frac{2 * \frac{c_i}{t_i} * \frac{c_i}{n_i}}{\frac{c_i}{t_i} + \frac{c_i}{n_i}}\right), \tag{3}$$

em que m é o número de questões avaliadas, c_i é o número de respostas corretas retornadas pelo modelo para a questão i, t_i é o número total de respostas retornadas pelo modelo para a questão i e n_i é o número de respostas corretas existentes para a questão i.

Já a métrica Micro *F1-score* é uma medida da acurácia do modelo considerando-se as respostas individualmente, independente dos conjuntos de respostas das questões. Assim ela é definida como,

$$Micro\ F1 = \left(\frac{2 * \frac{c_t}{t_t} * \frac{c_t}{n_t}}{\frac{c_t}{t_t} + \frac{c_t}{n_t}}\right), \tag{4}$$

em que c_t é o número de respostas corretas retornadas pelo modelo considerando-se todas as questões, t_t é o número total de respostas retornadas pelo modelo considerando-se todas as questões e n_t é o número de respostas corretas existentes considerando-se todas as questões.

A métrica *H@1*, denominada *Hits at* 1, é importante para verificar-se a confiabilidade da primeira resposta retornada pelo modelo para as questões e é definida como,

$$H@1 = \frac{c}{m}, \tag{5}$$

onde m é o número de questões avaliadas e c é o número de questões para as quais a primeira resposta retornada pelo modelo é uma resposta correta.

A métrica Acurácia Real é uma medida utilizada para verificar a capacidade dos modelos de acertarem completamente os conjuntos de respostas das questões. Assim, é definida como,

$$Acurácia\ Real = \frac{q}{m}, \tag{6}$$

onde m é o número de questões avaliadas e q é o número de questões para as quais o modelo é capaz de acertar todo o conjunto de respostas.

Por fim, para analisar a capacidade dos modelos de não retornarem respostas erradas para as questões, implementou-se uma métrica que aqui é denominada de Taxa de Questões com Apenas Verdadeiro Positivos (TP), definida como,

$$TP = 100 * \frac{q}{m}, \tag{7}$$

em que m é o número de questões avaliadas e q é o número de questões para as quais dentre o conjunto de respostas retornados pelo modelo para a questão, não haja nenhuma resposta errada.

3. RESULTADOS E DISCUSSÕES

De acordo com a metodologia aplicada, foram reavaliados vários modelos voltados para a tarefa de *Knowledge Base Question Answering (KB-QA)* que pudessem compor o *Ensemble*, com abordagens de recuperação de informação, de análise semântica ou de análise semântica neural. Diversas abordagens foram tomadas para estruturação do *Ensemble*, considerando-se diferentes conjuntos de modelos. Dentre as configurações abordadas, a que apresentou melhor resultado em termos de *F1-score* foi o *Ensemble* por voto majoritário composto pelos modelos *Multihop* (Lan and Jiang, 2020), *NS-CQA* (Hua et al., 2020) e o *BAMnet* (Chen et al., 2019), que foram melhores avaliados individualmente de acordo com a reavaliação realizada.

Os resultados de desempenho do *Ensemble* por voto majoritário e das reavaliações dos modelos selecionados com relação a *F1-score* são sintetizados na Tabela 2. Além disso, na Tabela 2 são apresentadas outras métricas auxiliares que permitem realizar uma análise mais ampla de vários aspectos do *Ensemble* que são importantes para sua implementação como parte de um assistente virtual.

Observando-se os resultados da Tabela 2 é possível notar que o *Ensemble* é capaz de superar todos os três modelos individuais no aspecto de *F1-score*, que é a principal métrica de cobertura das respostas pelos modelos (e principal métrica de desempenho adotada), bem como nas outras

Tabela 2. Desempenho dos modelos individuais selecionados para diferentes métricas relevantes

	Multihop	*NS-CQA*	*BAMnet*	*Ensemble*
F1-score	0,7312	0,7138	0,5722	0,7540
Precisão	0,7344	0,7130	0,6198	0,7703
Recall	0,7947	0,7391	0,6324	0,7803
Micro F1	0,5804	0,8509	0,2708	0,7661
Acurácia Real	0,6431	0,6425	0,3667	0,6480
H@1	0,7297	0,7175	0,6162	0,7657

métricas auxiliares, com exceção da Micro *F1-score* onde o modelo *NS-CQA* consegue um melhor resultado.

O *Ensemble* cujos resultados são descritos na Tabela 2, no entanto, possui brecha para aprimoramento uma vez que possui 192 questões onde não ocorre consenso dentre os modelos para as respostas. Seguindo como descrito na metodologia aplicada, por meio da aplicação do *Ensemble* simples ao conjunto de validação e reavaliação dos modelos individuais para as questões sem consenso do conjunto de validação, são gerados os resultados da Tabela 3.

Tabela 3. Desempenho dos modelos individuais selecionados para questões sem consenso do conjunto de validação

	Multihop	*NS-CQA*	*BAMnet*
F1-score	0,2800	0,4378	0,0012

Através da análise da Tabela 3 são desenvolvidos os modelos *Ensemble* com contramedida, onde a contramedida é o modelo *NS-CQA*, e o *Ensemble* com contramedida total, onde de acordo com a Tabela 3, as prioridades são *NS-CQA* > *Multihop* > *BAMnet* (neste caso, considerando-se o banco de dados *WebQuestionsSP*, o *BAMnet* acaba não fornecendo respostas uma vez que os casos restantes em que o *NS-CQA* não fornece respostas são completamente supridos pelo *Multihop*, no entanto, isto pode não acontecer em outros bancos de dados). Os resultados de desempenho para ambos os novos *Ensembles* gerados em relação a *F1-score* e outras métricas auxiliares, para o conjunto de teste, são mostrados na Tabela 4, onde CS significa contramedida simples e CT contramedida total.

Tabela 4. Desempenho dos modelos finais de *Ensembles* desenvolvidos para diferentes métricas relevantes

	Ensemble CS	*Ensemble* CT
F1-score	0,7872	0,8143
Precisão	0,8034	0,8311
Recall	0,8138	0,8456
Micro F1	0,8579	0,8420
Acurácia Real	0,6803	0,7010
H@1	0,7993	0,8255

Observando-se os resultados dispostos na Tabela 4 dos modelos de *Ensembles* com contramedida e com contramedida total em comparação aos resultados do modelo de *Ensemble* sem nenhum tipo de contramedida mostrado na Tabela 2 é possível verificar que utilizar-se contramedidas para os casos sem consenso garante uma melhora de desempenho para ambos os modelos (contramedida e contramedida total) em todos os aspectos das tabelas. Além disso, o modelo de *Ensemble* com contramedida total possui o desempenho em geral superior ao modelo com contramedida simples, com exceção do micro *F1-score*.

Com os resultados obtidos é possível destacar os seguintes benefícios visando o emprego dos *Ensembles* desenvolvidos no módulo *soft-KB lookup* proposto:

a) Considerando o *F1-score*: empregar o *Ensemble* com contramedida total garante uma maior cobertura em geral das respostas corretas das questões. É importante notar que o modelo *Multihop* é o modelo *KB-QA* estado-da-arte para o banco de dados *WebQuestionsSP* com um *F1-score* de 74,0% (Lan and Jiang, 2020) e o *Ensemble* com contramedida total atinge um *F1-score* de 81,43%;
b) Considerando-se a Precisão: o *Ensemble* com contramedida total garante que dentre as respostas retornadas pelo modelo haja um maior número de respostas corretas, com um aumento de precisão de 9,67% em relação ao modelo individual com a maior precisão (Multihop);
c) Considerando-se o *Recall*: o *Ensemble* com contramedida total atinge um aumento no *Recall* de 5,09% com relação ao melhor modelo individual neste aspecto (Multihop) o que garante que maior número das respostas corretas sejam recuperadas pelo modelo;
d) Considerando-se o Micro *F1-score*: neste caso o único modelo capaz de superar o melhor modelo individual foi o *Ensemble* com contramedida simples, obtendo um desempenho de 85,79% superando o melhor modelo individual (*NS-CQA*) em 0,7%. Ainda assim, o *Ensemble* com contramedida total obtém um desempenho competitivo de 84,20%. A Micro *F1-score* é uma medida de acurácia a nível de respostas individuais do modelo. É preciso observar que os valores mais elevados do *Ensemble* simples, do *Ensemble* com contramedida simples e do *NS-CQA* se dá devido ao número de questões as quais não são capazes de responder, e considerando-se isto em conjunto com as outras análises realizadas, verifica-se uma abrangência geral superior por parte do *Ensemble* com contramedida total;
e) Considerando-se a Acurácia Real: o *Ensemble* com contramedida total atinge uma Acurácia Real de 70,10%, o que representa uma melhora de desempenho de 5,31% em relação ao melhor modelo individual neste aspecto (Multihop), o que mostra que o modelo é capaz de acertar um maior número de questões completamente;
f) Considerando-se *H@1*: o *Ensemble* com contramedida total possui um desempenho de 82,55%, uma superação de 9,58% em relação ao melhor modelo individual nesta métrica (Multihop). Existem situações em que o usuário não necessariamente necessita de um conjunto de respostas completo mas apenas uma resposta correta, e nestes casos o modelo com maior *H@1* possui uma maior chance de suprir tal necessidade.

Para complementar a análise, na Tabela 5 são mostrados os resultados de desempenho obtidas pelos *Ensembles* gerados e pelos modelos individuais empregados, com relação a taxa de questões com apenas verdadeiros positivos (TP) como descrita na Subseção 2.4 e também com relação a taxa de questões com apenas verdadeiros positivos desconsiderando-se as questões sem respostas (TP*), onde S significa simples, CS contramedida simples e CT contramedida total.

Tabela 5. Desempenho dos modelos individuais selecionados e dos *Ensembles* com relação a métrica TP

Modelos	TP	TP*	#Questões sem respostas
Multihop	69,12%	69,12%	0
NS-CQA	93,41%	68,88%	402
BAMnet	48,62%	48,62%	0
Ensemble S	85,05%	73,34%	192
Ensemble CS	84,75%	76,63%	133
Ensemble CT	79,07%	79,07%	0

De acordo com os resultados dispostos na Tabela 5, o melhor desempenho com relação à métrica TP dentre modelos individuais e *Ensembles* é o *NS-CQA* com uma taxa de 93,41%, que supera o segundo melhor modelo (o *Ensemble* simples) por uma taxa de 8,36%. Dentre as três versões de *Ensembles* desenvolvidas, observa-se na Tabela 5 uma redução da TP a medida que são fornecidas respostas para mais questões.

Pensando-se por exemplo, em um assistente virtual voltado para atendimento de suporte em um *e-commerce*, um valor de TP mais alto traz uma maior garantia de que não será fornecida uma resposta errada ao cliente, o que poderia agravar o problema atual do cliente ou mesmo criar outro problema, o que seria inconveniente e poderia trazer prejuízos a empresa.

É importante notar, no entanto, que a maior taxa de questões com apenas verdadeiros positivos obtida pelo modelo *NS-CQA* se dá principalmente devido ao modelo não retornar respostas para uma grande quantidade de questões (24,53% das questões não são respondidas pelo modelo), o que também é inconveniente pensando-se no caso hipotético de um assistente virtual para atendimento de suporte em um *e-commerce*, uma vez que o assistente seria incapaz de solucionar boa parte dos problemas. Além disso, pensando em um assistente virtual voltado para entretenimento um modelo que não é capaz de responder a muitas questões acaba sendo vago e entediante, o que acaba afastando os usuários (Adiwardana et al., 2020).

Desta forma, uma maior taxa de questões com apenas verdadeiros positivos *TP* acompanhado de uma grande quantidade de questões não respondidas, no geral, não satisfaria o problema. Com isto, uma análise mais interessante pode ser avaliar os modelos com relação a taxa de questões com apenas verdadeiros positivos desconsiderando-se as questões sem respostas dos modelos (TP*) dispostas na Tabela 5. Com base na TP*, verifica-se que ambos os *Ensembles* conseguem aprimorar as taxas de questões com apenas verdadeiros positivos (com respostas) e que tomar contramedidas que permitam aos modelos que sejam capazes de responder a mais questões ainda os torna mais acurados neste sentido.

É notável que considerando-se a TP*, o melhor modelo de *Ensemble* desenvolvido (o *Ensemble* com contramedida total) consegue superar o melhor modelo individual nesta métrica (o *Multihop*) em 9,95%. Observa-se ainda que o modelo individual *NS-CQA* sofre uma redução abrupta de 24,53% considerando suas TP e TP* dispostas na Tabela 5, o que o torna inferior neste aspecto ao *Multihop* que era o segundo melhor modelo individual considerando-se a TP.

À luz dos resultados, análises e discussões apresentadas, supõe-se que a melhor situação seja conciliar maiores TP* e *F1-score*. Assim, apesar de não possuir a mais alta TP, o *Ensemble* com contramedida total é capaz de agregar muitos benefícios em relação aos modelos individuais avaliados, o que o torna a melhor opção em geral para no futuro ser empregado como parte de um Assistente Virtual.

4. CONCLUSÃO

Com a análise dos diferentes modelos para a tarefa de *Knowledge Base Question Answering (KB-QA)* e o desenvolvimento dos *Ensembles* por voto majoritário simples, com contramedida e com contramedida total, atingiu-se um resultado satisfatório, antes ainda não obtido na literatura, com um incremento na *F1-score* de 7,43% em relação ao modelo *KB-QA* individual estado-da-arte, considerando-se o banco de dados *WebQuestionsSP*. Neste sentido, aplicar tal modelo como parte de um assistente virtual poderia apresentar resultados interessantes.

Para trabalhos futuros, é possível tentar aprimorar ainda mais o modelo combinado via *Ensemble* buscando-se novos modelos para *KB-QA* individuais, em vista de que visando os 100% em F1-score, o modelo *Ensemble* com contramedida total ainda tem uma margem de aproximadamente 19% para melhora, além de que o modelo individual *BAMnet* empregado nos *Ensembles* é relativamente inferior aos outros dois modelos individuais empregados, com uma diferença em *F1-score* superior a 14%, o que tende a inferiorizar o resultado final na média, havendo um certo nível de diversidade de respostas dentre os modelos. Pode-se também aplicar diferentes configurações de *Ensembles*, empregando-se mais modelos, ou empregando-se informações das saídas dos modelos individuais e informações de classificação das questões para seleção de respostas, à nível individual ou de conjunto.

Por fim, em trabalhos futuros, visando o emprego de tais modelos como parte de um assistente virtual voltado para o *e-commerce*, é necessário o levantamento de bancos de dados que sejam interessantes e voltados para o comércio eletrônico, e a realização de análises e testes dos modelos para este nicho específico. No mais, são necessários o desenvolvimento dos outros módulos para um assistente virtual, para gerenciar o diálogo, gerar respostas em linguagem natural e lidar com outas características do diálogo, como opiniões e emoções apresentadas pelos usuários.

AGRADECIMENTOS

Agradecimento especial à Omnilogic Inteligência S/A, FAPEMIG, CNPq e Universidade Federal de Lavras.

REFERÊNCIAS

ABCOMM (2021). O crescimento dos marketplaces em 2021. URL https://abcomm.org/noticias/o-crescimento-dos-marketplaces-em-2021/.

Adiwardana, D., Luong, M.T., So, D.R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., and Le, Q.V. (2020). Towards a Human-like Open-Domain Chatbot. *arXiv preprint arXiv:2001.09977*. URL http://arxiv.org/abs/2001.09977.

Chen, Y., Wu, L., and Zaki, M.J. (2019). Bidirectional attentive memory networks for question answering over knowledge bases. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2913–2923. Association for Computational Linguistics, Minneapolis, Minnesota. doi:10.18653/v1/N19-1299. URL https://www.aclweb.org/anthology/N19-1299.

Fu, B., Qiu, Y., Tang, C., Li, Y., Yu, H., and Sun, J. (2020). A survey on complex question answering over knowledge base: Recent advances and challenges. *CoRR*, abs/2007.13069. URL https://arxiv.org/abs/2007.13069.

Gao, J., Galley, M., and Li, L. (2018). Neural approaches to conversational AI. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Tutorial Abstracts*, 2–7. doi:10.18653/v1/p18-5002.

He, G., Lan, Y., Jiang, J., Zhao, W.X., and Wen, J.R. (2021). Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, 553–561. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3437963.3441753. URL https://doi.org/10.1145/3437963.3441753.

Hu, S., Zou, L., Yu, J.X., Wang, H., and Zhao, D. (2017). Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 30(5), 824–837.

Hua, Y., Li, Y.F., Qi, G., Wu, W., Zhang, J., and Qi, D. (2020). Less is more: Data-efficient complex question answering over knowledge bases. *Journal of Web Semantics*, 65, 100612. doi:https://doi.org/10.1016/j.websem.2020.100612. URL https://www.sciencedirect.com/science/article/pii/S1570826820300470.

Lan, Y. and Jiang, J. (2020). Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 969–974. Association for Computational Linguistics, Online. doi:10.18653/v1/2020.acl-main.91. URL https://aclanthology.org/2020.acl-main.91.

Ochieng, P. (2020). Parot: Translating natural language to sparql. *Expert Systems with Applications: X*, 5, 100024. doi:https://doi.org/10.1016/j.eswax.2020.100024. URL https://www.sciencedirect.com/science/article/pii/S2590188520300032.

Yih, S.W.t., Chang, M.W., He, X., and Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*.

Zhou, L., Gao, J., Li, D., and Shum, H.Y. (2020). The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *Computational Linguistics*, 46(1), 53–93. doi:10.1162/coli_a_00368. URL https://doi.org/10.1162/coli_a_00368.