Universidade Federal de São Carlos– UFSCar
Centro de Ciências Exatas e de Tecnologia– CCET
Departamento de Computação– DC
Programa de Pós-Graduação em Ciência da Computação– PPGCC

João Gabriel Melo Barbirato

Construção automática de grafo de conhecimento no domínio do e-commerce

São Carlos
2022

João Gabriel Melo Barbirato

Construção automática de grafo de conhecimento no domínio do e-commerce

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Metodologias e Técnicas de Computação

Orientador: Profa. Dra. Helena de Medeiros Caseli

São Carlos
2022

FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO - PPGCC/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
Telefone: (16) 33518233 - http://www.ufscar.br

DECLARAÇÃO

Declaro, para os devidos fins, que a banca de defesa de dissertação de mestrado do(a) aluno(a) João Gabriel Melo Barbirato intitulada “Construção automática de grafo de conhecimento no domínio do e-commerce”, foi realizada no dia 27 de abril de 2022 às 14h00min, remotamente, e o(a) mesmo(a) foi Aprovado(a). Declaro ainda que os seguintes membros participaram da comissão examinadora: Helena de Medeiros Caseli (Presidente - PPGCC - UFSCar), Daniela Barreiro Claro (UFBA) e Ivanovitch Medeiros Dantas da Silva (UFRN).

O(a) aluno(a) está apto(a) agora a receber o título de “Mestre em Ciência da Computação”. A homologação do título e o respectivo processo de expedição e registro de diploma aguardam a finalização da versão final do texto da dissertação e de seu depósito no Repositório Institucional da UFSCar para se iniciarem.

Documento assinado eletronicamente por **Ivan Rogerio da Silva**, **Assistente em Administração**, em 04/05/2022, às 11:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufscar.br/autenticacao, informando o código verificador **0647914** e o código CRC **E5BBC68D**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.009668/2022-74 SEI nº 0647914

Modelo de Documento: Declaração, versão de 02/Agosto/2019

Para você, 2 anos no futuro.

Agradecimentos

Agradecimentos ao Departamento de Computação da UFSCar. Agradeço também à empresa parceira americanas s.a., a qual incentiva e fomenta essa e outras pesquisas na UFSCar. Em especial à parceria no projeto de extensão "Dos dados ao conhecimento: extração e representação de informação no domínio do e-commerce" (Projeto de extensão #23112.000186/2020-97), do qual este trabalho faz parte. Agradecimentos especiais aos professores Daniel Lucrédio e Diego Furtado, os quais são membros dessa parceria também.

"Chega mais perto e contempla as palavras.
Cada uma
tem mil faces secretas sob a face neutra
e te pergunta, sem interesse pela resposta,
pobre ou terrível, que lhe deres:
Trouxeste a chave?"
(Carlos Drummond de Andrade)

Resumo

Extrair conhecimento de forma eficiente, quando grandes volumes de dados são gerados diariamente, ainda é um desafio. Na maioria dos casos esses dados são não estruturados, ou seja, são apresentados no formato textual ou visual sem a clara delimitação das informações que contém e das relações entre essas informações. Assim, tão importante quanto extrair corretamente o conhecimento é representá-lo e armazená-lo de modo que ele seja útil. Uma das formas de representar (armazenar) esse conhecimento é por meio de grafos de conhecimento. Essas estruturas representam relações semânticas (arestas) entre entidades (vértices), como a relação semântica `é_um` entre as entidades `maçã` e `fruta` representada pela tripla: `é_um(maçã,fruta)`. Assim, este trabalho aborda a construção automática de um grafo de conhecimento para o domínio do e-commerce, onde os vértices desse grafo representam produtos e caraterísticas, enquanto as arestas conectando esses vértices são usadas para descrever a relação entre eles. Entre os desafios que este trabalho enfrentou está o de ter de lidar com dados não estruturados, ruidosos e incompletos gerados pelos usuários no domínio do e-commerce. A esse fato somam-se os desafios semânticos do domínio, uma vez que os dados do e-commerce carregam mais valor semântico por se tratarem de entidades reais e de categorias e contextos muito variados. Com o intuito de avançar na investigação de métodos para lidar com tais desafios e peculiaridades do domínio do e-commerce, neste trabalho foram treinados dois modelos de grafo para a recomendação de produtos: um deles seguindo métodos distributivos através da ferramenta RedisGraph, e outro que explora propriedades latentes dos métodos distribuídos de *embeddings* de grafo de conhecimento. Os resultados mostram que o último pode contribuir para tarefas no domínio do e-commerce que visam a diversidade de produtos.

Palavras-chave: Processamento de Língua Natural. Representação de Conhecimento. Grafo de Conhecimento. E-commerce.

Abstract

Extracting knowledge efficiently, when large volumes of data are generated daily, is still a challenge. In most cases, these data are unstructured, that is, they are presented in textual or visual format without a clear delimitation of the information they contain and the relationships between this information. Thus, as important as correctly extracting knowledge is to represent it and store it so that it is useful. One of the ways to represent (store) this knowledge is through knowledge graphs. These structures represent semantic relationships (edges) between entities (vertices), as the semantic relationship `is_a` between the `apple` and `fruit` entities represented by the triple: `is_a(apple,fruit)`. Thus, this work addresses the automatic construction of a knowledge graph for the e-commerce domain, where the vertices of this graph represent products and characteristics, while the edges connecting these vertices are used to describe the relationship between them. Among the challenges that this work faced is having to deal with unstructured, noisy and incomplete data generated by users in the e-commerce domain. Added to this fact are the semantic challenges of the domain, since e-commerce data carry more semantic value because they are real entities that came from very varied categories and contexts. In order to advance in the investigation of methods to deal with such challenges and peculiarities of the e-commerce domain, in this work two graph models were trained for product recommendation: one of them following distributive approach through the RedisGraph tool, and another that explores latent properties of the distributed methods of knowledge graph embeddings. The results show that the latter can contribute to tasks in the e-commerce domain that aim at product diversity.

Keywords: Natural Language Processing. Knowledge Representation. Knowledge Graph. E-commerce.

Lista de ilustrações

Lista de tabelas

Lista de siglas

BEM *Bayes Embedding*

BERT *Bidirectional Encoder Representations from Transformers*

CE classificação de entidades

CPV *Category-Property-Value*

CT classificação de triplas

KGE *Embeddings* de Grafo de Conhecimento

FN Falsos Negativos

FP Falsos Positivos

KG Grafo de Conhecimento

BG Grafo de Comportamento

GNN *Graph Neural Network*

PKG Grafo de produtos

KB Base de Conhecimento

KGC Completude de Grafos de Conhecimento

MTB *matching the blanks*

MR *Mean Rank*

MRR *Mean Reciprocal Rank*

F1 *Medida F*

MLM modelo de linguagem mascarada

mBERT Multilingual BERT

REN Reconhecedor de Entidade Nomeada

PL predição de links

PLN Processamento de Língua Natural

RE resolução de entidades

RW random walk

CNN Rede Neural Convolucional

SPO *Subject-Property-Object*

SP *shortest path*

VP Verdadeiros Positivos

Sumário

Capítulo 1

Introdução

A produção constante de dados está presente no cotidiano do século XXI. Como evidencia Domo (2018), a cada dia, cerca de $2,5$ quintilhões de dados são produzidos, ou seja, aproximadamente $1,7$MB por segundo. Além disso, Domo (2018) também estima que cerca de $3,5$ EB[1] serão gerados por humanos até 2025.

Sabendo que o hábito de comprar online também cresce, parte desses dados são produzidos por plataformas do e-commerce. No Brasil, em 2020, cerca de $42,9$ milhões de pessoas consumiram produtos em plataformas do e-commerce (DIGITAIS, 2021). Ainda em 2020, segundo Medeiros (2020), $90,8$ milhões de produtos foram vendidos nessas plataformas. Além disso, somente em março de 2022, as plataformas de e-commerce brasileiro registraram $1,68$ bilhões de visitas (RODRIGUES, 2022). Evidentemente, essas vendas geram grandes volumes de dados diariamente, e, com isso, surge a necessidade de se extrair conhecimento sobre eles.

Contudo, extrair conhecimento de forma eficiente, quando grandes volumes de dados são gerados diariamente, ainda é um desafio. Na maioria dos casos esses dados são não estruturados, ou seja, são apresentados no formato textual ou visual sem a clara delimitação das informações que contém e das relações entre essas informações. Assim, tão importante quanto extrair corretamente o conhecimento é representá-lo ou armazená-lo de modo que ele seja útil.

Nesse contexto, os dados não estruturados textuais são compostos por palavras que combinadas formam sentenças. Em uma sentença em língua natural cada palavra desempenha um papel, de tal forma que sua posição, suas informações morfossintáticas e seu significado no mundo são relevantes para o entendimento geral do conteúdo que aquela sentença representa. A forma como o contexto de ocorrência das palavras afeta o sen-

[1] *Exabytes*, ou seja, da ordem de 10^{18} *bytes*.

tido de uma palavra ou sentença é objeto de estudo de diversos trabalhos focados na representação textual.

Na literatura, a representação de conhecimento textual aponta duas abordagens distintas: a representação distributiva, a qual consiste em representar cada palavra com base na sua disponibilidade no córpus, como a abordagem *bag-of-words*; e a representação distribuída, capaz de representar palavras de acordo com possíveis relações entre elas derivadas com base na similaridade de contextos de ocorrência, como as *word embeddings* (MIKOLOV et al., 2013). Essa última é mais sofisticada e escalável, ao passo que se trata de uma representação de informação mais densa. Para ilustrar a diferença de informação armazenada/recuperada usando cada uma dessas representações a Tabela 1 traz exemplos dos 5 vizinhos mais próximos recuperados a partir da consulta do termo `motorola` usando TF-ID e *word embeddings* para o mesmo conjunto de dados. Dos termos recuperados por *word embddings*, três são modelos de produtos da marca Motorola (`xt`, `chocolight` e `xtmoto`). Entretanto, apenas dois dos recuperados pelo TF-IDF mostram-se relevantes: um atributo (`cor`), possivelmente associado a um item da marca, e um valor para ele (`preto`).

Tabela 1 – Recuperação de termos relacionados a `motorola`.

TF-IDF			*word embeddings*		
Recuperado	Ranque	*Score*	Recuperado	Ranque	*Score*
`cor`	1	1,000	`xt`	1	1,000
`sao`	2	0,971	`chocolight`	2	0,628
`preto`	3	0,792	`samsumg`	3	0,624
`informacoes`	4	0,779	`xtmoto`	4	0,506
`meramente`	5	0,755	`motorolacompativel`	5	0,322

No domínio do e-commerce, as abordagens de Recuperação de Informação mais utilizadas para plataformas são baseadas em representação distributiva, como *bag-of-words* e TF-IDF. No entanto, esses métodos estão fadados a recuperar documentos irrelevantes para a busca do usuário (KUTIYANAWALA; VERMA et al., 2018). Por exemplo, quando se procura por `TV Samsung` nessas plataformas, provavelmente também será recuperado o item indesejado `controle remoto para TV Samsung`.

Contrapondo essa limitação, uma das formas distribuídas de representar (armazenar) conhecimento é por meio de **grafos de conhecimento** (KG, do inglês, *Knowledge Graph*), estruturas bastante adotadas, principalmente, pela eficiência e eficácia no modo como organizam a informação (YAN et al., 2018).

Os KGs são constituídos por **entidades** e **relações** entre elas. As entidades são representadas como **vértices** (ou nós) no grafo, e as relações (binárias) entre entidades são indicadas pelas **arestas**. Dessa forma, duas entidades (representadas em vértices) estão relacionadas se houver uma aresta conectando-as. O objetivo principal dessas estruturas é representar conhecimento de forma organizada e compreensível (YAN et al., 2018).

Essas estruturas semânticas poderosas são utilizadas, por exemplo, para resolver problemas de Processamento de Língua Natural (PLN), como citado por Wang et al. (2017): na extração de relações (MINTZ et al., 2009; RIEDEL; YAO; MCCALLUM, 2010; WESTON et al., 2013; JIANG et al., 2016) e em sistemas de perguntas e respostas (BORDES; WESTON; USUNIER, 2014; BORDES; CHOPRA; WESTON, 2014).

Entretanto, é necessário extrair relações para construir um KG. Essa tarefa consiste em identificar fatos em dados não estruturados, como textos em língua natural (WANG et al., 2017). Segundo Oliveira et al. (2015), a extração e o uso de relações semânticas é essencial para o entendimento automático de dados textuais. Por exemplo, na sentença `"iPhone X é um dos Smartphones mais vendidos na atualidade"`, o extrator de relações pode predizer que existe uma relação de hiponímia (`is_a` ou `é_um`) entre a entidades `iPhone` e `Smartphone`.

Contudo, uma das limitações na construção de grandes grafos de conhecimento é a necessidade de algum trabalho manual. Sabou et al. (2005) apontam que é necessário partir de bases de referência, criadas por especialistas em um determinado domínio. Além disso, principalmente quando se trata de textos, uma mesma palavra pode incorporar diversos sentidos e, com isso, as relações que a contemplam precisam ser estabelecidas por decisões abertas à interpretação (OLIVEIRA et al., 2015). Jiang et al. (2019) apontam que KGs de domínio sempre serão considerados incompletos, por conta da construção manual do conhecimento que os popula.

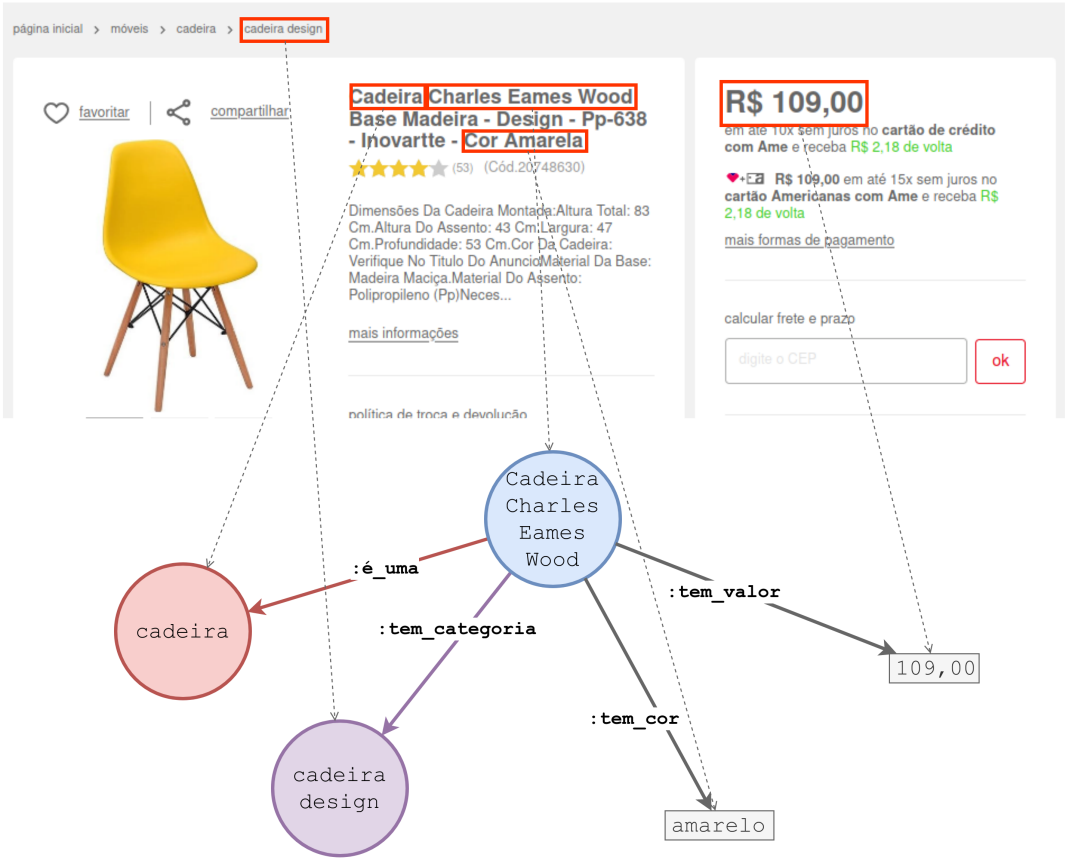
No domínio do e-commerce não é diferente. Apesar do uso de KG no domínio do e-commerce estar sendo estudado extensivamente na atualidade (por exemplo, nos trabalhos do capítulo 3), sua aplicação nesse domínio não é uma tarefa trivial.

Para ilustrar peculiaridades desse domínio considere, por exemplo, o título do produto, sua foto, descrição e outras informações estruturadas apresentadas na Figura 1 . Diversas relações semânticas estão presentes nesse exemplo e seriam facilmente identificadas por um humano como: `(Cadeira Charles Eames Wood, tem_categoria, cadeira design)`, `(Cadeira Charles Eames Wood, tem_cor, amarelo)`, `(Cadeira Charles Eames Wood, tem_valor, 109,00)`, entre outras.

Essas informações podem ser organizadas em um KG como o ilustrado na Figura 1. Embora a identificação dessas informações (entidades, conceitos, valores) e relações entre elas seja trivial para um humano, o mesmo não ocorre no processamento automático que envolveria processamento textual (e às vezes visual), desambiguação de conceitos e conhecimento de mundo para gerar o KG. A essas dificuldades do processamento automático somam-se outras específicas do domínio do e-commerce, apontadas por Xu et al. (2020):

- **Falta de confiabilidade nos dados** – Os modelos de KG de domínio geral trabalham com fatos bastante verossímeis, enquanto que dados extraídos no domínio do e-commerce muitas vezes são baseados nas interações entre usuários e produtos. Por essa origem, a informação nesse domínio está sujeita à erros humanos.

Figura 1 – Exemplo de informações de um produto a venda no site da Americanas.



Fontes: próprio autor <https://www.americanas.com.br/>. Último acesso: 25 de agosto de 2020.

- ❑ **Riqueza semântica dos dados** – Relações entre produtos são semanticamente mais ricas, por se tratarem de objetos no mundo real. Em outras palavras, quando uma entidade está envolvida em uma relação `tem_valor` há conhecimento implícito que tal entidade é um objeto que pode ser comprado. Esse conhecimento de mundo implícito torna o processamento superficial do texto inadequado em muitos casos.

Apesar desses desafios, o uso de KG no domínio do e-commerce é promissor. Por exemplo, KGs mostraram-se úteis e produziram melhores resultados em sistemas de recomendação (YU et al., 2014; YE et al., 2019; ZHANG et al., 2020) e busca (JIANG et al., 2019; ZHANG et al., 2020; XU et al., 2020), principalmente por permitirem interações entre informações geradas por usuários e relações entre produtos (ZHANG et al., 2016). Além disso, estudam-se diferentes estratégias de como modelar dados do e-commerce e utilizá-los para oferecer melhores experiências aos clientes (YE et al., 2019; JIANG et al., 2019; XU et al., 2020; ZHANG et al., 2020).

Nesse contexto, métodos de *Embeddings* de Grafo de Conhecimento (KGE) se destacam por representarem entidades e relações de forma distribuída. As *Embeddings* de Grafo de conhecimento são representações vetoriais de elementos de um KG no espaço. Por

meio dessas representações, é possível analisar entidades e relações através de operações algébricas. Por conta da natureza latente das *embeddings*, essas operações permitem inferir fatos implícitos sobre esses elementos, como, por exemplo, a possível similaridade não esperada entre uma mesa digitalizadora e uma câmera. Embora métodos distributivos de recuperação de informação, como o Solr[2], sejam populares, a literatura evidencia suas limitações (KUTIYANAWALA; VERMA et al., 2018). Por isso, este trabalho propõe o uso de métodos KGE nas tarefas finais no domínio do e-commerce.

1.1 Objetivo e hipóteses

Neste contexto, este trabalho visa investigar a construção automática de grafos de conhecimento no domínio do e-commerce a partir de dados textuais não estruturados escritos em português do Brasil.

Para tanto foi utilizada uma base textual de produtos contendo informações como títulos, descrições e *queries* de busca, fornecidas pela empresa parceira deste projeto: a Americanas S.A.. Trata-se de uma companhia digital dona de várias marcas do e-commerce brasileiro como: Americanas[3], Submarino[4], Shoptime[5] e Soub![6].

Nesse contexto, este projeto visa responder a seguinte questão de pesquisa:

QP Métodos KGE são efetivos na construção automática de um KG no domínio do e-commerce a partir de dados não estruturados em português do Brasil?

Para responder a QP, será avaliado se Grafos de Conhecimento gerados como resultado deste projeto são efetivos, ou seja, podem auxiliar em algum dos problemas do e-commerce. Neste trabalho, aborda-se a tarefa de recomendação de produtos. Nesse contexto, a **hipótese** deste trabalho é a de que técnicas de representação distribuídas (implementadas no KGE) serão mais efetivas do que as distributivas, uma vez que tendem a encontrar termos ligados à relações semânticas entre entidades e não apenas associados à co-ocorrência de palavras.

Além da grande quantidade de dados não estruturados presentes na base da Americanas S.A., alguns produtos também possuem dados estruturados em suas fichas técnicas. Esses dados estruturados podem ser usados para comparar com a quantidade de informação relacional que é possível extrair de dados não estruturados.

[2] https://solr.apache.org/
[3] <https://www.americanas.com.br/>
[4] <https://www.submarino.com.br/>
[5] <https://www.shoptime.com.br/>
[6] <https://www.soubarato.com.br/>

1.2 Organização da monografia

Este texto está organizado como segue. No Capítulo 2 são descritos os principais conceitos relacionados à construção de KG. Em seguida, o Capítulo 3 traz um apanhado geral das abordagens de construção de KG no domínio geral e no domínio do e-commerce. O Capítulo 4 descreve os recursos utilizados e os dois métodos de construção de KGs investigados neste trabalho. No Capítulo 5 são apresentados os resultados obtidos para a recomendação de produtos utilizando grafos. Por fim, o Capítulo 6 evidencia as conclusões deste trabalho, bem como sugere trabalhos futuros.

Capítulo 2

Construção automática de grafos de conhecimento

Neste capítulo serão apresentados: os conceitos referentes aos grafos de conhecimento e *Embeddings* de Grafo de Conhecimento (KGE) (seção 2.1), um apanhado das principais abordagens na construção automática de KGE (seção 2.2) e quais são as principais medidas automáticas usadas na avaliação da qualidade dessas estruturas (seção 2.3).

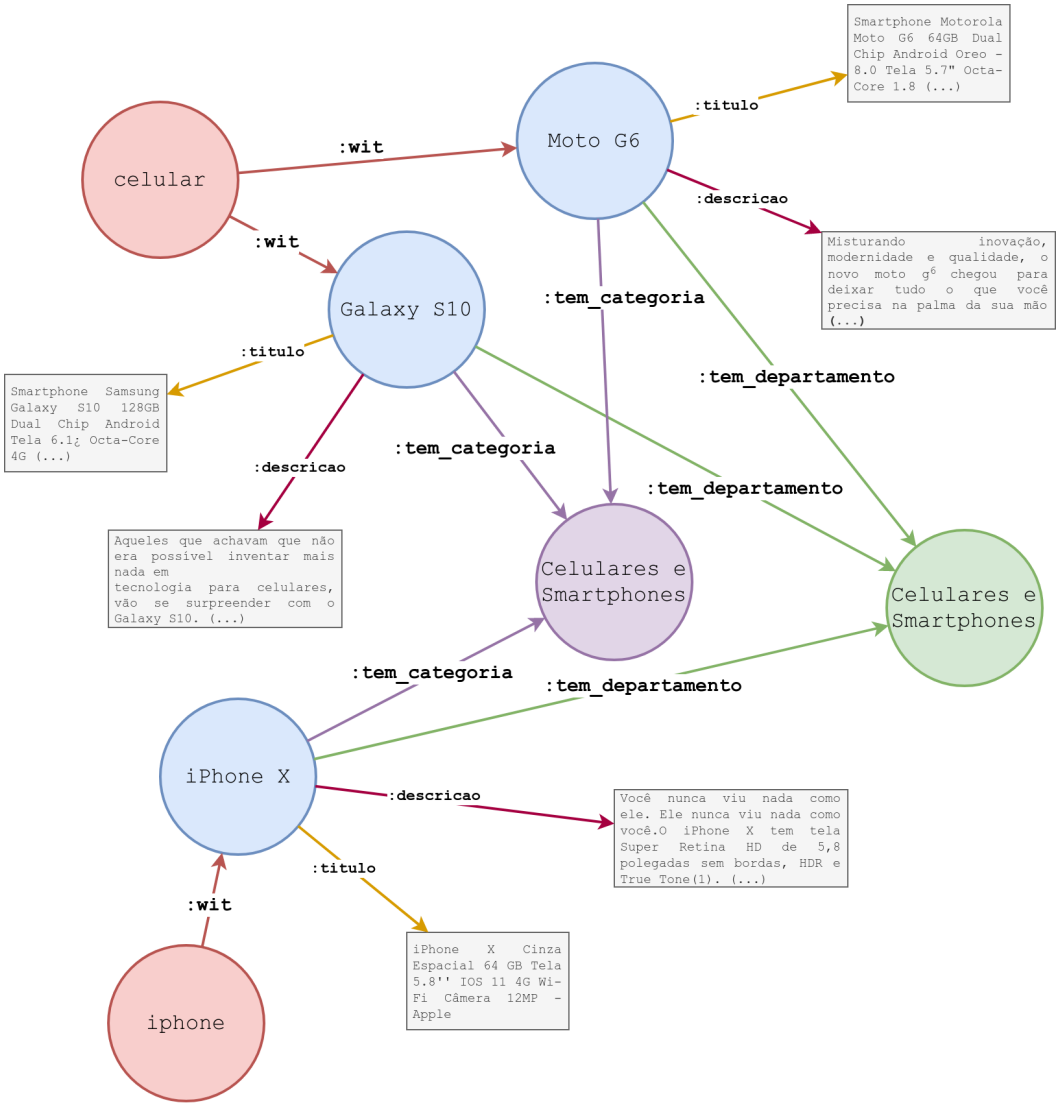
2.1 Grafos de conhecimento

Grafos de Conhecimento (KGs) são estruturas semânticas compostas por **vértices** – ou nós – e **arestas**, de forma que os vértices representam conceitos – ou entidades – e arestas, relações semânticas entre conceitos (YAN et al., 2018).

Formalmente, sejam E e R conjuntos de entidades e relações, respectivamente. Um grafo de conhecimento é uma coleção de triplas (h, r, t) (ou triplas *Subject-Property-Object* (SPO)), onde $h, t \in E$ e $r \in R$ (SUN et al., 2019). Cada tripla (h, r, t) pode ser considerada como um fato. No grafo presente na Figura 2, é possível encontrar, por exemplo, a relação `wit`[1] (r) entre `celular` (h) e `Galaxy S10` (t), formando a tripla (`celular`, `wit`, `Galaxy S10`).

[1] A empresa parceira define como *What Is This* (WIT) uma classificação específica de produtos que engloba características suficientemente distintas de outras WITs, como `celular` é a WIT de `Galaxy S10`. Existe uma hierarquia de WITs e um produto pode pertencer à mais de uma WIT: uma cadeira pode pertencer à WIT `sala de estar` e à WIT `cadeira`, por exemplo.

Figura 2 – Exemplo de grafo de conhecimento fictício instanciado para o domínio do e-commerce. Vértices de mesma cor representam um mesmo tipo de entidade, assim como arestas de mesma cor se referem a um mesmo tipo de relação.



Fonte: próprio autor

2.1.1 *Embeddings* de Grafo de Conhecimento (KGE)

Muito embora a estrutura de grafo seja eficiente para representar dados estruturados, sua natureza simbólica pode ser difícil de se manipular (WANG et al., 2017). Assim, as abordagens mais recentes (BORDES et al., 2013b; WANG et al., 2014; JI et al., 2015a; JI et al., 2015b; NICKEL; TRESP; KRIEGEL, 2011; YANG et al., 2015; NICKEL et al., 2016; TROUILLON et al., 2016; DETTMERS et al., 2018; NGUYEN et al., 2018) utilizam *embeddings*[2] para representar entidades e relações em um espaço vetorial no que chamam de *Knowledge Graph Embeddings* (ou KGE).

[2] *Embeddings* ou *word embeddings* (MIKOLOV et al., 2013) são representações vetoriais de palavras. Esses vetores são construídos com base na premissa de preservar o significado latente da palavra. Assim, é possível preservar conceitos e manipulá-los matematicamente. Por exemplo, no espaço de *embedding*, tem-se $\texttt{Paris} - \texttt{França} \approx \texttt{Londres} - \texttt{Inglaterra}$.

A ideia geral da construção de *embeddings* de grafos de conhecimento considera os seguintes fatores:

1. **Espaço de *embedding*** – tem-se uma base de dados povoada de fatos (triplas) (h, r, t). Cada entidade h, t é representada no espaço vetorial como uma *embedding* de baixa dimensão. Entretanto, uma relação r pode ser representada tanto diretamente como uma *embedding* no mesmo espaço de h e t (BORDES et al., 2013b; YANG et al., 2015; NICKEL et al., 2016; TROUILLON et al., 2016; DUVENAUD et al., 2015; DETTMERS et al., 2018; NGUYEN et al., 2018) quanto como uma *embedding* intermediada por outra forma de representação, por exemplo, hiperplanos (WANG et al., 2014) e matrizes (NICKEL; TRESP; KRIEGEL, 2011; JI et al., 2015a; JI et al., 2015b; JI et al., 2016).

2. **Função *score*** – cada relação r possui uma função $f_r(h, t)$ que mede a relevância da tripla candidata (h, r, t). Essa função depende da abordagem adotada para a geração do modelo[3]. A literatura mostra trabalhos que calculam *score*, por exemplo, baseando-se em distância entre entidades (BORDES et al., 2013b; WANG et al., 2014; JI et al., 2015a; JI et al., 2015b), *matching* semântico (NICKEL; TRESP; KRIEGEL, 2011; SOCHER et al., 2012; YANG et al., 2015; NICKEL et al., 2016; TROUILLON et al., 2016) e redes neurais convolucionais (DUVENAUD et al., 2015; DETTMERS et al., 2018; NGUYEN et al., 2018).

3. **Triplas negativas** – para treinar o modelo de predição de *links*, são geradas triplas (h', r, t) ou (h, r, t'), consideradas triplas negativas ou corrompidas. Assim, o treinamento favorece as triplas positivas em detrimento das negativas (BORDES et al., 2013b), o que ajuda a evitar *overfitting*.

4. **Otimização** – a otimização é baseada na suposição de que o valor *score* de uma tripla candidata seja maior do que uma tripla corrompida (SUN et al., 2019).

Muitos estudos apontam que grafos de conhecimento são incompletos por natureza (JIANG et al., 2019; XU et al., 2020), ou seja, sempre é possível encontrar novas instâncias de relações ou desambiguar entidades do grafo. Nesse contexto, diversos trabalhos visam resolver o problema da Completude de Grafos de Conhecimento (KGC). Três tarefas principais estão relacionadas a esse problema, são elas:

- ❑ Predição de links (PL) – é a tarefa de predizer qual entidade possui uma determinada relação com outra entidade, ou seja, predizer a entidade h, dados r e t, ou t, dados r e h. A tarefa provou-se fundamental para o aprendizado de KGs utilizando *embeddings*, por aprender relações latentes entre entidades. Na prática, trata-se

[3] Modelo, no contexto de KGE, é uma instância do grafo, ou seja, um conjunto de *embeddings* gerado seguindo um método.

de realizar um cálculo entre entidades e relações no espaço de *embeddings*, o que é simples e eficiente (ZHANG et al., 2020).

- ❏ Classificação de triplas (CT) – consiste em classificar se uma tripla (h, r, t) nova (que não havia sido aprendida até então) é verdadeira ou falsa. Triplas com valores de *score* maiores tendem a ser fatos verdadeiros e, para isso, utilizam-se limiares específicos δ_r para cada relação. Se a função de *score* $f_r(h, t) \geq \delta_r$, entende-se que o fato é verdadeiro e, portanto, a nova tripla deve ser inserida no KG.

- ❏ Classificação de entidades (CE) – essa tarefa visa atribuir categorias às entidades. Na Figura 2, por exemplo, os nós `iPhone X`, `Galaxy S10` e `Moto G6` pertencem à mesma categoria.

- ❏ Resolução de entidades (RE) – Além de incompletos, os grafos de conhecimento podem conter informações redundantes. Para isso, a tarefa de resolução de entidades consiste em identificar duas entidades que se referem ao mesmo objeto.

Dessa forma, a construção de KGs se mostra promissora em diversos domínios do conhecimento, por ser versátil em representar informações do mundo real. Sobretudo, o uso de *embeddings* em KGs mostra-se promissor no domínio do e-commerce (JIANG et al., 2019; YE et al., 2019; SANH; WOLF; RUDER, 2019; ZHANG et al., 2020). Neste trabalho utilizou-se a tarefa de PL para treinar os modelos KGEs, os quais estão melhor detalhados na seção 4.3.2.2.

2.2 Principais abordagens para geração de KGE

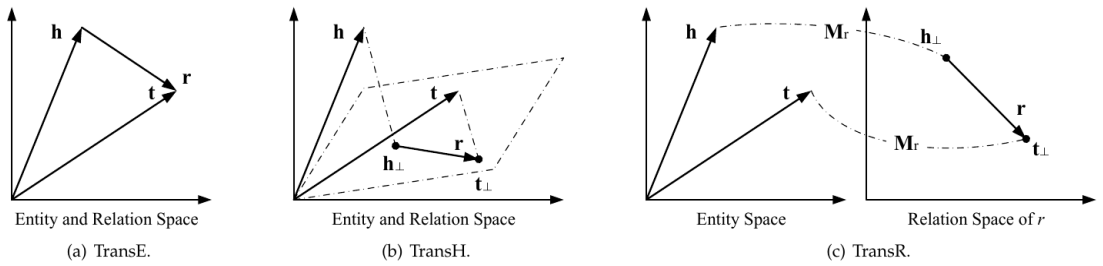
Diversos métodos para geração de KGEs foram propostos. Esses métodos podem ser agrupados, de acordo com a função de *score* que utilizam, como: (1) métodos de translação, (2) métodos de *matching* semântico e (3) métodos de redes neurais convolucionais.

No entanto, esses métodos utilizam estratégias de treinamento muito diferentes, o que dificulta as comparações entre eles. Ruffinelli, Broscheit e Gemulla (2020) estudam a performance de diversos métodos sumarizando e quantificando empiricamente o impacto de cada estratégia. Assim, percebe-se que, por exemplo, o modelo RESCAL (NICKEL; TRESP; KRIEGEL, 2011), um dos principais e mais diretos modelos baseados em *matching* semântico, teria uma performance comparável ao modelo ConvE (DETTMERS et al., 2018), um dos modelos mais recentes baseados em redes neurais convolucionais, quando ambos usam estratégias semelhantes. Por isso, neste documento, não se identifica um conjunto de modelos estado-da-arte para a área de KGE como um todo.

2.2.1 Métodos de translação

Métodos de translação baseiam-se na distância entre *embeddings* de entidades h e t para verificar a verossimilhança[4] do fato (h, r, t). Essa distância é medida após uma operação de translação baseada na relação r. A Figura 3 ilustra essa ideia.

Figura 3 – Ilustração espacial do funcionamento dos modelos de KGE baseados em translação: (a) TransE, (b) TransH e (c) TransR.



Fonte: (WANG et al., 2017)

O modelo de translação mais representativo é o **TransE** (BORDES et al., 2013b), o qual baseia-se no pressuposto de que se duas entidades de um fato (h e t) estão ligadas pela relação (r), então $h + r \approx t$. Trata-se de uma ideia simples e eficaz. Por exemplo, no contexto de *embeddings*, é intuitivo pensar que `GracilianoRamos` + `Escreveu` ≈ `VidasSecas`.

Contudo, o TransE não identifica relações 1-para-N, N-para-1 ou N-para-M; apenas relações 1-para-1, o que inviabiliza também considerar como válido `GracilianoRamos` + `Escreveu` ≈ `Caetés`. A partir dessa limitação, novas propostas surgiram para estudar translações dependentes de uma relação, para que `VidasSecas` e `Caetés` sejam semelhantes considerando-se a relação `Escreveu` e possivelmente distantes considerando-se as demais relações.

O **TransH** (WANG et al., 2014) segue essa ideia, tratando relações não mais como vetores, mas como hiperplanos no espaço de *embedding*. Assim, as projeções ortogonais $h_\perp, t_\perp$ de duas entidades h e t no hiperplano cujo vetor normal é w_r são interligadas por r; ou seja, o modelo assume que $h_\perp + r \approx t_\perp$ Desse modo, o TransH consegue lidar com relações N-para-M com N ≥ 1 e M ≥ 1. No entanto, ainda persiste um problema de ambiguidade, uma vez que relações ressaltam diferentes aspectos para diferentes entidades. Por exemplo, o termo "grande" tratando-se de vestidos é diferente de "grande" no contexto de eletrodomésticos.

Para lidar com esse problema surgiu o **TransR** (JI et al., 2015a), que ressalta esses diferentes aspectos nas relações. Esse modelo pressupõe que relações transformam *entidades* para outro espaço vetorial de embedding. Assim, para a matriz de transformação M_r, as entidades projetadas $h_\perp = M_r h$ e $t_\perp = M_r t$ são interligadas por r; ou seja, semelhantemente ao modelo TransH, o TransR assume que $h_\perp + r \approx t_\perp$. Porém, essa abordagem

[4] Um fato (h, r, t) é verossímil se é muito provável que exista uma relação r que interligue h e t, respectivamente, como sujeito e objeto da relação.

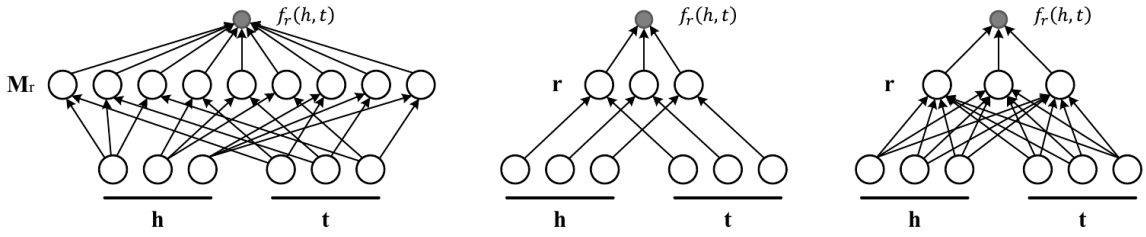
foge da simplicidade dos modelos TransE e TransH. Muito embora o TransR seja eficiente em modelar relações semanticamente complexas, ao introduzir matrizes específicas para cada relação, a quantidade de parâmetros aumenta consideravelmente.

O **TransD** (JI et al., 2015b) propõe, então, diminuir a complexidade de parâmetros decompondo a matriz M_r, proposta no TransR, em dois vetores w_h e w_t dependentes de qual "posição" uma entidade se encontra num determinado fato (h, r, t), ou ainda, a qual categoria cada entidade pertence; o que, na prática, calcula M_r de forma dinâmica. No entanto, essa abordagem desfavorece relações heterogêneas e desbalanceadas, como a relação de `gênero`, a qual liga diversas entidades de pessoas a poucas entidades de valores possíveis, como `masculino` e `feminino`. Assim, o **TranSparse** (JI et al., 2016) propõe lidar com essas relações utilizando matrizes de transformação esparsas M_r^h e M_r^t. Dessa forma, é possível reconhecer relações desbalanceadas, ao custo do aumento de parâmetros para o treinamento do modelo.

2.2.2 Métodos de *matching* semântico

Por outro lado, os métodos de *Matching* Semântico se baseiam na similaridade para verificar a verossimilhança de um fato. Esses utilizam aspectos semânticos latentes nas *embeddings* de entidades e relações. A Figura 4 ilustra essa ideia.

Figura 4 – Ilustração espacial do funcionamento dos principais métodos de KGE baseados em *matching* semântico: (a) RESCAL, (b) DistMult e (c) HolE.



(a) RESCAL. (b) DistMult. (c) HolE.

Fonte: (WANG et al., 2017)

O **RESCAL** (NICKEL; TRESP; KRIEGEL, 2011), ou modelo bilinear, associa cada entidade h, t com um vetor no espaço de *embedding* e cada relação com uma matriz. Dessa forma, a função de *score* utiliza as interações entre todos os pares de entidades e é calculada como $f_r = h^\top M_r t^\top$. Trata-se de um método eficiente. Contudo, esse método traz complexidade quadrática para cada relação analisada. Por isso, o método **DistMult** (YANG et al., 2015) simplifica o RESCAL tornando M_r uma matriz diagonal, calculando-se a função *score* como um produto interno: $f_r =< h, diag(M_r), t >$.

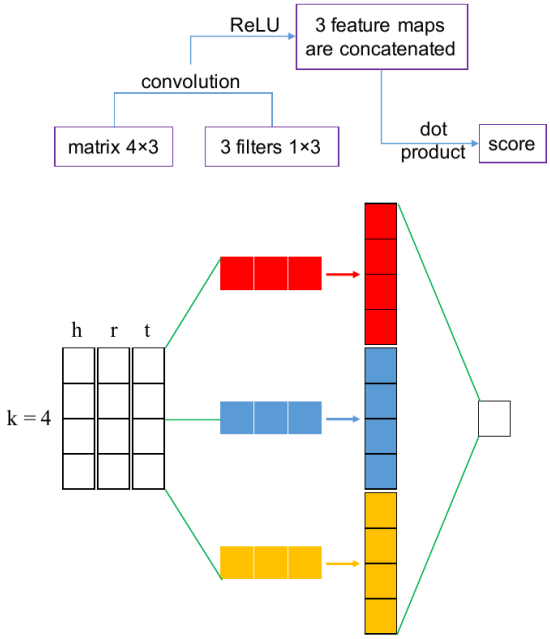
Porém, por se tratar de um produto interno, a diagonalização favorece relações simétricas em detrimento das assimétricas. Assim, o **HolE** (NICKEL et al., 2016) propõe utilizar a correlação circular $h \star t$ para a função de *score* $f_r = r^\top (h \star t)$. Já que $h \star t \neq t \star h$,

é possível identificar relações assimétricas. Nessa ideia, o método **ComplEx** (TROUILLON et al., 2016) também propõe tratar relações assimétricas, mas utilizando o produto interno no espaço dos números complexos[5] $\mathbb{C}$, de modo que $f_r = Re(< h, r, \overline{t} >)$, onde $Re(z)$ representa a parte real do número complexo z, uma vez que $< h, r, \overline{t} > \neq < \overline{h}, r, t >$.

2.2.3 Métodos de redes neurais convolucionais

Por fim, com o advento do aprendizado profundo, foram propostos métodos que extraem características semânticas de entidades e relações por meio de redes neurais convolucionais.

Figura 5 – Ilustração do funcionamento do ConvKB.



Fonte: (NGUYEN et al., 2018)

A rede **R-GCN** (DUVENAUD et al., 2015) propõe, dadas as *embeddings* das entidades de uma tripla (h, r, t), concatenar convoluções em um único vetor de representação. Esse vetor, então, é decodificado pelo método DistMult (YANG et al., 2015).

O **ConvE** (DETTMERS et al., 2018), por outro lado, propõe que a convolução aconteça com as *embeddings* da relação e da entidade sujeito (primeiro elemento da tripla), resultando em uma concatenação na forma de uma matriz bidimensional. A justificativa é que, dessa forma, é possível extrair características mais significativas.

Por fim, o **ConvKB** (NGUYEN et al., 2018) propõe que a matriz seja composta, também, da *embedding* da entidade objeto (último elemento da tripla). Dessa forma, o

[5] Um número complexo $z \in \mathbb{C}$ pode ser escrito da forma $z = x_1 + x_2 i$ tal que $i^2 = -1$ é a unidade imaginária e $x_1, x_2 \in \mathbb{R}$ são, respectivamente, a parte real e a parte imaginária do número z. Além disso, o conjugado do número z é dado como $\overline{z} = x_1 - x_2 i$.

ConvKB consegue, em geral, melhores resultados por generalizar características transicionais, ou seja, computar relações onde uma entidade possa ser tanto sujeito como objeto, como na relação `pai` em (`João`, `pai`, `Felipe`) e (`Antônio`, `pai`, `João`). A Figura 5 ilustra, além do funcionamento do ConvKB (NGUYEN et al., 2018), como os métodos utilizam redes neurais convolucionais em geral. Na parte de cima da figura é possível ter uma visão generalizada da aplicação de redes neurais convolucionais para KGE: há uma estrutura de entrada composta pelas *embeddings* de uma tripla (h, r, t); em seguida, aplicam-se filtros de convolução formando uma estrutura intermediária; e, por fim, há uma função (nesse caso, a função ReLU[6]) que transforma a estrutura intermediária em um valor de *score*.

2.3 Avaliação

Para Sabou et al. (2005), um sistema automático de construção de grafos de conhecimento deve ser avaliado em duas perspectivas diferentes: (1) deve ser capaz de aprender corretamente, ou seja, de aprender conceitos corretos e (2) deve ser capaz de aprender conceitos novos, ou seja, conceitos que ainda não foram aprendidos ou não estão no KG base.

Diversas medidas para a avaliação intrínseca da construção automática de grafos de conhecimento foram propostas na literatura. Bordes et al. (2013b) propõem o uso das medidas baseadas em *ranking*, descritas na seção 2.3.1: (2.3.1.1) *Hits@N*, (2.3.1.2), *Mean Rank* (MR) e (2.3.1.3) *Mean Reciprocal Rank* (MRR).

Após a construção do KG, avalia-se de forma extrínseca também o seu desempenho em tarefas finais. Assim, a literatura evidencia o uso de medidas convencionais de classificação – (2.3.2.1) precisão, (2.3.2.2) revocação e (2.3.2.3) medida F – para tarefas como classificação de triplas, classificação de entidades e classificação de relações.

2.3.1 Avaliação intrínseca

Esta seção descreve medidas que avaliam a completude e a relevância das triplas aprendidas num KG, como na tarefa de PL. Assim, como exemplo para as seções 2.3.1.1, 2.3.1.2 e 2.3.1.3, considere as triplas positivas (`gato`, `come`, `atum`) e (`cachorro`, `emite`, `latido`) pertencentes a T_p e os respectivos conjuntos ranqueados de triplas (T_i) geradas para $h_1 =$ `gato` e $r_1 =$ `come` e $h_2 =$ `cachorro` e $r_2 =$ `emite`, apresentadas respectivamente nas Tabelas 2 e 3.

[6] ReLU (*Rectifier Linear Unit*) é uma função de ativação comumente utilizada em aprendizado profundo e é definida como $f(x) = max(0, x)$ onde x é a entrada de um neurônio.

Tabela 2 – Conjunto de triplas contendo a tripla positiva (`gato`, `come`, `atum`), em negrito.

Tripla	Score
$t_{1,2}$ (gato,come,latido)	0,95
$t_{1,1}$ **(gato,come,atum)**	0,75
$t_{1,3}$ (gato,come,miado)	0,60
$t_{1,4}$ (gato,come,pular)	0,55

Tabela 3 – Conjunto de triplas contendo a tripla positiva (`cachorro`, `emite`, `latido`), em negrito.

Tripla	Score
$t_{2,1}$ **(cachorro,emite,latido)**	0,95
$t_{2,3}$ (cachorro,emite,carne)	0,50
$t_{2,2}$ (cachorro,emite,lamber)	0,30
$t_{2,4}$ (cachorro,emite,atum)	0,20

2.3.1.1 *Hits @ k*

Hits@k é o número de vezes que uma tripla positiva t_i aparece no conjunto $T_i^{(k)}$, de tamanho k e cujos elementos são ordenados decrescentemente por uma função *score* para cada exemplo de triplas positivas, como define a Equação 1.

$$hits@k = \frac{1}{|T_p|}\sum_{i=1}^{|T_p|}|T_p \cap T_i^{(k)}| \tag{1}$$

Considerando o exemplo no início da seção 2.3.1, tem-se:

$$hits@1 = \frac{1}{|\{t_{1,1}, t_{2,1}\}|}\left[\left|T_p \cap \{t_{1,2}\}\right| + \left|T_p \cap \{t_{2,1}\}\right|\right] = \frac{1}{2}[|\emptyset| + |\{t_{2,1}\}|] = \mathbf{0{,}5}$$

$$hits@2 = \frac{1}{|\{t_{1,1}, t_{2,1}\}|}\left[\left|T_p \cap \{t_{1,2}, t_{1,1}\}\right| + \left|T_p \cap \{t_{2,1}, t_{2,3}\}\right|\right] = \frac{1}{2}[|\{t_{1,1}\}| + |\{t_{2,1}\}|] = \mathbf{1}$$

Valores tradicionalmente usados para k são 10 e 5 em *hits@10* e *hits@5*. Além disso, os valores dessa medida variam entre 0 e 1, 0 e quanto maior, melhor.

2.3.1.2 *Mean Rank* (MR)

Seja $rank(t, T)$ o *score* ordenado decrescentemente) de uma tripla t no conjunto T. Então, a medida *Mean Rank* (MR) de um conjunto de triplas é a média dos *rankings* das triplas positivas entre seus conjuntos gerados, como define a Equação 2.

$$MR = \frac{1}{|T_p|}\sum_{t_i \in T_p} rank(t_i, T_i) \tag{2}$$

Considerando o exemplo no início da seção 2.3.1, tem-se:

$$MR = \frac{1}{|\{t_{1,1}, t_{2,1}\}|}\left[rank(t_{1,1}, \{t_{1,2}, t_{1,1}, t_{1,3}, t_{1,4}\}) + rank(t_{2,1}, \{t_{2,1}, t_{2,3}, t_{2,2}, t_{2,4}\}\right]$$

$$MR = \frac{1}{2}\Big[2 + 1\Big] = \mathbf{1{,}5}$$

Os valores dessa medida iniciam em 1 e o limite superior depende do tamanho do conjunto em avaliação. Quanto menor o valor, melhor.

2.3.1.3 *Mean Reciprocal Rank* (MRR)

Trata-se de uma medida semelhante à MR (definida em 2.3.1.2), contudo utilizando o inverso dos *rankings*, como mostra a Equação 3.

$$MRR = \frac{1}{|T_p|} \sum_{t_i \in T_p} \frac{1}{rank(t_i, T_i)} \tag{3}$$

Considerando o exemplo no início da seção 2.3.1 e de forma análoga à seção 2, tem-se:

$$MRR = \frac{1}{2}\left[\frac{1}{2} + 1\right] = \mathbf{0{,}75}$$

Os valores dessa medida variam de 0 a 1,0 e quanto maior, melhor.

Neste trabalho, foram utilizadas as medidas *Hits @ k* e MR para avaliar o treinamento de modelos KGE.

2.3.2 Avaliação extrínseca

Esta seção descreve as medidas utilizadas para avaliar a eficiência do KG em aplicações finais, como extração automática de relações, CT e CE. Dessa forma, para as medidas presentes em 2.3.2.1, 2.3.2.2 e 2.3.2.3, considera-se para um determinado rótulo:

- ***Verdadeiros Positivos (VP)*** – quantidade de exemplos do rótulo de interesse preditos corretamente;
- ***Falsos Negativos (FN)*** – quantidade de exemplos do rótulo de interesse preditos incorretamente;
- ***Falsos Positivos (FP)*** – quantidade de exemplos preditos incorretamente como o rótulo de interesse.

2.3.2.1 Precisão

A precisão visa medir quantos exemplos que foram encontrados são relevantes, e é definida com a Equação 4.

$$precisão = \frac{VP}{VP + FP} \tag{4}$$

2.3.2.2 Revocação

A revocação visa medir quantos exemplos relevantes foram encontrados, e é definida com a Equação 5.

$$revocação = \frac{VP}{VP + FN} \tag{5}$$

2.3.2.3 *Medida F* (F1)

Trata-se de uma combinação entre precisão e revocação e pode ser definida com a Equação 6.

$$medida\text{-}f = \frac{2 \times precisão \times revocação}{precisão + revocação} \tag{6}$$

2.3.2.4 Problema Multiclasse

Quando o problema de predição de relações envolve mais de duas classes, consideram-se soluções para classificação não-binária, ou multiclasse. Assim, sendo n o total de classes únicas, define-se a média micro de uma medida como descreve a Equação 7. É possível notar que a média micro favorece classes mais frequentes.

$$medida\ média_{micro} = \frac{1}{n}\sum_{i=1}^{n} medida_i \tag{7}$$

Além disso, as médias macro para precisão, revocação e F1 podem ser definidas respectivamente conforme as Equações 8, 9, 10. Evidentemente, a medida individual de cada classe possui a mesma relevância no cálculo da média macro.

$$precisão\ média_{macro} = \frac{\sum_{i=1}^{n} VP_i}{\sum_{i=1}^{n} VP_i + FP_i} \tag{8}$$

$$revocação\ média_{macro} = \frac{\sum_{i=1}^{n} VP_i}{\sum_{i=1}^{n} VP_i + FN_i} \tag{9}$$

$$f1\ média_{macro} = \frac{\sum_{i=1}^{n} 2 \times precisão_{macro} \times revocação_{macro}}{\sum_{i=1}^{n} precisão_{macro} + revocação_{macro}} \tag{10}$$

Neste trabalho, a medida F1 micro foi utilizada na seção 4.3.1.1 para eleger o modelo BERT com melhor performance na tarefa proposta.

Capítulo 3

Trabalhos Relacionados

Com o intuito de encontrar os trabalhos mais relevantes para este projeto, buscaram-se trabalhos relacionados com o auxílio da ferramenta *Parsif.al*[1]. Foram encontrados 115 trabalhos não duplicados coletados dos principais repositórios de publicações científicas em computação: *ACM Digital Library*[2], *IEEE Digital Library*[3], *ACL Anthology*[4] e *Springer Link*[5].

Para isso, utilizou-se a seguinte *string* de busca:

("e-commerce"OR "Portuguese") AND ("Automatic Learning"OR "Ontology Learning"OR "Knowledge Base Completion"OR "Knowledge Base Embedding"OR "Knowledge Graph Completion"OR "Knowledge Graph Embedding"OR "Ontology Construction"OR "Semantic Relations") AND ("Ontology"OR "Knowledge Base"OR "Knowledge Graph")

Dos 115 trabalhos selecionados, 13 foram aceitos. Desses, foram escolhidos os 4 mais relevantes para o projeto, os quais estão descritos em detalhes na seção 3.2. Além desses, na seção 3.1 são descritos também dois métodos de construção de *embeddings* de grafos de conhecimento no domínio geral e um recentemente proposto, baseado no BERT. Em suma, a seção 3.3 resume as principais informações sobre os métodos descritos neste capítulo.

[1] <https://parsif.al/about/>
[2] <https://dl.acm.org/>
[3] <ieeexplore.ieee.org/>
[4] <https://www.aclweb.org/anthology/>
[5] <https://link.springer.com/>

3.1 Abordagens no domínio geral

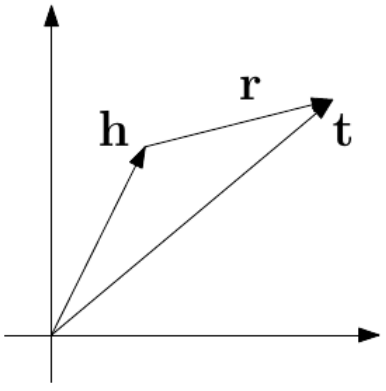
Esta seção apresenta os métodos selecionados como os mais relevantes para tratar o problema de construção automática de um KG no domínio geral. Para tanto, três métodos de construção automática de KGs no domínio geral serão descritos, dos quais dois deles (3.1.1 e 3.1.2) são usados em vários trabalhos de construção de KGs.

3.1.1 TransE (BORDES et al., 2013b)

Bordes et al. (2013b) apresentam o TransE, um método de KGE, que se baseia na suposição de que relações podem ser representadas como translações no espaço vetorial. Essa suposição diz que, sendo h e t duas entidades e r uma relação entre elas, então a *embedding* para a entidade t deve estar próxima à *embedding* da entidade h somada a algum vetor que depende da relação r. Esse método gera *embeddings* para entidades e relações e, portanto, entidades e relações compartilham um mesmo espaço vetorial.

Assim, sendo h e t duas *embeddings* de entidades, o modelo define a *embeddidng* r da relação entre elas como $r = h - t$, como ilustrado na Figura 6. Desse modo, o método ranqueia a relação como a distância $d(h + r, t) = ||h + r - t||_2^2$. Por exemplo, a relação `bornIn`, partindo da entidade `LilWayne` levaria em `NewOrleans` porque a entidade `NewOrleans` foi melhor ranqueada para essa relação, partindo dessa primeira entidade, do que as entidades `Atlanta` e `Toronto`. Dessa forma, o contraste entre triplas positivas e negativas, na fase de treinamento do modelo, consiste em otimizar a diferença entre o ranking da tripla positiva $d(h + r, t)$ e o da negativa $d(h + r, t')$.

Figura 6 – Exemplo do método de translação TransE.



Fonte: (WANG et al., 2014)

O treinamento ocorre como ilustrado no algoritmo 1. Dado um conjunto de treinamento composto por triplas $(h, r, t,)$ de entidades h e t e relação r, o método aprende as *embeddings* de entidades e de relações. O conjunto de exemplos negativos (*corrupted triplet* no algoritmo) é composto por triplas de treinamento nas quais uma ou outra entidade (h ou t) é substituída por uma entidade aleatória, mas não as duas ao mesmo tempo.

Algoritmo 1 Aprendizado TransE

Require: Conjuntos de Treino T, Entidades E, Relações R; Margem γ; Dimensão k;
1: $R \leftarrow$ **distr_unif**$(R, (-\frac{6}{\sqrt{k}}, -\frac{6}{\sqrt{k}}))$
2: $r \leftarrow ||r||$ **for** $r \in R$
3: $E \leftarrow$ **distr_unif**$(E, (-\frac{6}{\sqrt{k}}, -\frac{6}{\sqrt{k}}))$
4: **while** condicionais de parada não forem satisfeitas **do**
5: $e \leftarrow ||e||$ **for** $e \in E$
6: $T_{batch} \leftarrow$ **amostragem**(T, b) $\{$*minibatch de tamanho b*$\}$
7: $P_{batch} \leftarrow \emptyset$ $\{$*inicializa conjunto de pares de triplas*$\}$
8: **for** $(h, r, t) \in T$ **do**
9: $(h', r, t') \leftarrow$ **escolhe_tripla_negativa**$(T_{batch}, (h, r, t))$
10: $P_{batch} \leftarrow P_{batch} \cup \{((h, r, t), (h', r, t'))\}$
11: **end for**
12: **atualize_embedding**$(\sum_{((h,r,t),(h',r,t')) \in P_{batch}} \nabla[\gamma + d(h + r, t) - d(h' + r, t')]_+)$
13: **end while**

Fonte: Adaptada de (BORDES et al., 2013b)

Para o treinamento do modelo, Bordes et al. (2013b) utilizaram as bases Wordnet (MILLER, 1995) e Freebase (BOLLACKER et al., 2008) nas seguintes configurações: FB15k – base derivada da Freebase contendo 14.951 entidades e 1.345 relações, resultando em 592.213 triplas; WN18 – base derivada da WordNet contendo 40.943 entidades e 18 relações, resultando em 151.442 triplas; e a base de larga escala FB1M – derivada da Freebase, contendo 1 milhão de entidades que mais ocorriam e cerca de 23 mil relações, resultando em mais de 17 milhões de triplas.

Para a avaliação, utilizaram as medidas *hits@k* e *mean rank*, descritas respectivamente nas seções 2.3.1.1 e 2.3.1.2. O TransE foi o método de melhor desempenho entre todos os utilizados na comparação. Entre os resultados obtidos, vale mencionar o valor de $89,2$ em *hits@10* na base WN18, contra, por exemplo, $52,8$ obtido pelo modelo RESCAL (NICKEL; TRESP; KRIEGEL, 2011) e o *mean rank* de 251 do TransE contra 985 do RESCAL nessa mesma base.

Em geral, trata-se de uma abordagem simples, eficiente em termos de complexidade e número de parâmetros, competitiva e que serviu de *baseline* para diversos trabalhos da área de KGE na literatura. Bordes et al. (2013b) consolidaram bases de dados e medidas de avaliação como *benchmarks* usados em diversos trabalhos seguintes na área.

Contudo, como apontado por trabalhos como os de Wang et al. (2014) e Ji et al. (2015a), o modelo não resolve relações $1:N$, $N:1$ ou $N:N$. Por exemplo, a relação `directorOf`, partindo da entidade `AlfredHitchcock`, levaria em `Psycho`, `Rebecca` e `RearWindow`, todos esses sendo representados espacialmente pela mesma *embedding*.

Por fim, existem diversas implementações disponíveis para o TransE, entre elas o *toolkit* OpenKE[6] para geração de KGE pelo TransE e outros métodos conceituados na

[6] <https://github.com/thunlp/OpenKE>

literatura.

3.1.2 ComplEx (TROUILLON et al., 2016)

Trouillon et al. (2016) mostram um método de KGE baseado em *matching* semântico que utiliza multiplicação de vetores, diferentemente do método TransE (BORDES et al., 2013b), que utiliza translação. O ComplEx adota o produto interno Hermitiano entre as *embeddings* de um fato (h, r, t) para o cálculo da função *score*, operação essa que ocorre no espaço dos números complexos $\mathbb{C}$[7] (daí o nome, ComplEx).

Assim, o método tem a capacidade de considerar relações assimétricas para a função *score*. Essa assimetria se refere ao fato de que a entidade sujeito (h) de uma relação r_1 pode ser a entidade objeto (t) de uma relação r_2 e, portanto, essa assimetria deve ser considerada no cálculo. Um exemplo disso está nas relações de hierarquia, como em (`avô`, `maisVelho`, `pai`) e (`pai`, `maisVelho`, `filho`), onde a entidade "pai", na primeira tripla, ocupa a posição de objeto (t) e, na segunda tripla, a posição de sujeito (h).

Bordes et al. (2013a) afirmam que um modelo relacional deve ser capaz de aprender relações que, entre outras propriedades, sejam assimétricas. Contudo, os métodos de *matching* semântico desenvolvidos até aquela época não tinham essa capacidade pois se baseavam no produto interno do domínio real. Por outro lado, o produto interno Hermitiano não é simétrico, como o do domínio real $(\mathbb{R})$ e, ao usá-lo, Trouillon et al. (2016) mostraram que ele pode aprimorar a performance de tarefas envolvendo KGE.

Para isso, adotam a seguinte função *score*:

$$f_r(h, t) = Re(< w_r, h, \bar{t} >) \tag{11}$$

onde w_r é o vetor complexo que representa a relação r, h e t são *embeddings* de entidades, $\bar{t}$ é o conjugado da *embedding* t e $Re(z)$ é a parte real de um número complexo z. Por construção, a função da Equação 11 abrange relações reflexivas, irreflexivas, transitivas, simétricas e assimétricas; propriedades essas que Bordes et al. (2013a) descrevem como fundamentais para modelos relacionais.

Para avaliar o modelo, Trouillon et al. (2016) utilizaram três bases de dados: (1) uma sintética, constituída por 30 entidades e 2 relações (uma simétrica e a outra não), resultando em um total de 1.740 triplas; (2) a FB15k (BOLLACKER et al., 2008), contendo 14.951 entidades e 1.345 relações; e (3) a WN18 (MILLER, 1995), contendo 40.943 entidades e 18 relações.

O ComplEx se mostrou promissor na base sintética, superando os resultados do DistMult (YANG et al., 2015) e do TransE (BORDES et al., 2013b), mostrando que é capaz de assimilar informação assimétrica em uma base de dados pequena.

[7] Um número complexo $z \in \mathbb{C}$ pode ser escrito da forma $z = x_1 + x_2 i$ tal que $i^2 = -1$ é a unidade imaginária e $x_1, x_2 \in \mathbb{R}$ são, respectivamente, a parte real e a parte imaginária do número z. Além disso, o conjugado do número z é definido como $\overline{z} = x_1 - x_2 i$.

O mesmo acontece com o teste na FB15k, na qual obteve MRR (descrito em 2.3.1.3) de $0,692$ e hits@1 (descrito em 2.3.1.1) de $0,599$ contra $0,524$ e $0,402$, respectivamente, do HolE (NICKEL et al., 2016). Na WN18, entretanto, o Complex obteve uma performance mais evidente contra o DistMult (YANG et al., 2015) e o TransE (BORDES et al., 2013b), mas páreo ao HolE (NICKEL et al., 2016), obtendo MRR de $0,941$ contra $0,938$ do HolE. Isso ocorre principalmente devido à quantidade de relações assimétricas encontradas na base WN18. A Tabela 4 mostra os resultados de todos os métodos em comparação.

Tabela 4 – Resultados obtidos pelo método ComplEx na tarefa de predição de *links*.

Complex Embeddings for Simple Link Prediction

	WN18					**FB15K**				
	MRR		Hits at			MRR		Hits at		
Model	Filter	Raw	1	3	10	Filter	Raw	1	3	10
CP	0.075	0.058	0.049	0.080	0.125	0.326	0.152	0.219	0.376	0.532
TransE	0.454	0.335	0.089	0.823	0.934	0.380	0.221	0.231	0.472	0.641
DistMult	0.822	0.532	0.728	0.914	0.936	0.654	**0.242**	0.546	0.733	0.824
HolE*	0.938	**0.616**	0.93	**0.945**	**0.949**	0.524	0.232	0.402	0.613	0.739
ComplEx	**0.941**	0.587	**0.936**	**0.945**	0.947	**0.692**	**0.242**	**0.599**	**0.759**	**0.840**

Fonte: (TROUILLON et al., 2016)

A Tabela 5 ilustra o desempenho dos métodos ComplEx, Distmult e TransE para algumas relações da WN18. Na Tabela, é possível perceber que para as relações assimétricas, como `hypernym` (hiperonímia), `hyponym` (hiponímia), `part_of` (parte de) e `has_part` (tem parte), o ComplEx obteve resultados significativamente melhores que DistMult e TransE.

Tabela 5 – Desempenho dos métodos ComplEx, DistMult e TransE, medido em MRR.

Relation name	ComplEx	DistMult	TransE
hypernym	**0.953**	0.791	0.446
hyponym	**0.946**	0.710	0.361
member_meronym	**0.921**	0.704	0.418
member_holonym	**0.946**	0.740	0.465
instance_hypernym	**0.965**	0.943	0.961
instance_hyponym	**0.945**	0.940	0.745
has_part	**0.933**	0.753	0.426
part_of	**0.940**	0.867	0.455
member_of_domain_topic	**0.924**	0.914	0.861
synset_domain_topic_of	**0.930**	0.919	0.917
member_of_domain_usage	**0.917**	**0.917**	0.875
synset_domain_usage_of	**1.000**	**1.000**	**1.000**
member_of_domain_region	**0.865**	0.635	**0.865**
synset_domain_region_of	0.919	0.888	**0.986**
derivationally_related_form	**0.946**	0.940	0.384
similar_to	**1.000**	**1.000**	0.244
verb_group	**0.936**	0.897	0.323
also_see	0.603	**0.607**	0.279

Fonte: (TROUILLON et al., 2016)

Por fim, vale ressaltar que a implementação do método e seus *baselines* se encontram disponíveis abertamente em <https://github.com/ttrouill/complex>.

3.1.3 BERT para extração de relações (SOARES et al., 2019)

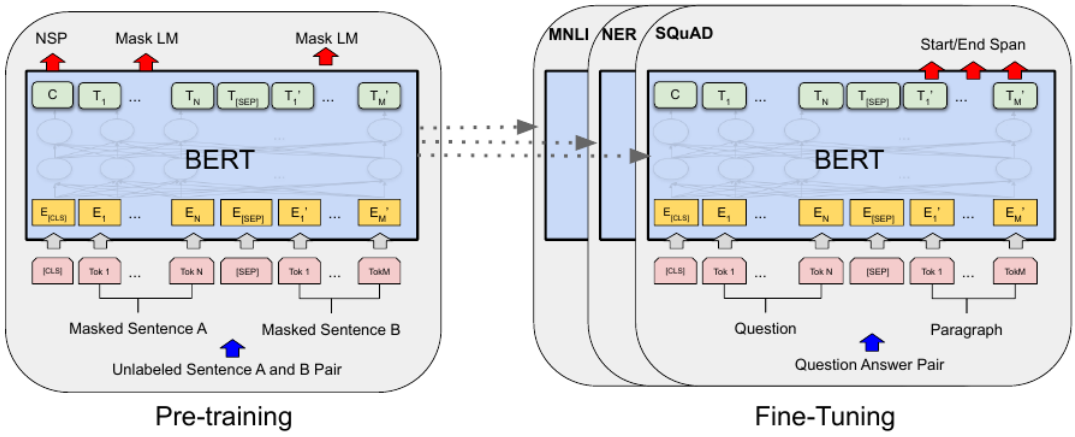
O trabalho de (SOARES et al., 2019) consiste em utilizar o BERT para a representação de relações via treinamento seguindo a estratégia de *matching the blanks* (MTB).

O *Bidirectional Encoder Representations from Transformers* (BERT) (DEVLIN et al., 2019) é uma arquitetura de encoder para geração de modelos de linguagem contextualizados. O modelo é versátil, capaz de entender contexto à esquerda e à direita para resolver diversas tarefas de PLN, como *Next Sentence Prediction*, *Question Answering* e *Sentiment Analysis* (DEVLIN et al., 2019).

A rede neural foi desenvolvida para adaptar as arquiteturas de modelagem de linguagem à tarefas em nível de sentença (DEVLIN et al., 2019). Diferentemente de seus predecessores ELMo (PETERS et al., 2018) e GPT (RADFORD et al., 2018), a capacidade do BERT de utilizar um modelo de linguagem mascarada (MLM) para os dados de entrada e, consequentemente, o treinamento de um transformador bidirecional faz com que a arquitetura seja mais versátil.

O treinamento de um modelo BERT se dá em duas etapas, como ilustra a Figura 7: o pré-treino, à esquerda, que consiste em treinar o modelo em diferentes tarefas não-supervisionadas; e o *fine-tuning*, que ajusta os parâmetros para uma determinada tarefa final (*downstream task*), como a Figura 7 mostra para a tarefa de *Question Answering* à direita.

Figura 7 – Funcionamento do modelo BERT.



Fonte: (DEVLIN et al., 2019)

Aplicando o BERT à tarefa de extração de relações binárias entre entidades, Soares et al. (2019) partem de um córpus de blocos de texto contendo duas entidades marcadas ligadas por uma relação. A partir disso, o conjunto de treino é criado substituindo a

entidade por um símbolo especial `[BLANK]` com o objetivo de prever a entidade ocultada. O símbolo é introduzido probabilisticamente para garantir que o modelo aprenda a relação não só pelas entidades, mas pelas palavras ao redor delas. Esse processo foi denomidado de *matching the blanks*. Para os autores, o treinamento com MTB visa resolver o problema de redundância observado nos dados em textos na web, onde um par de entidades arbitrário é provavelmente mencionado várias vezes ao longo de uma sequência.

Os autores propõem um método de representação denominado ENTITY MARKERS: dada uma sequência de tokens, iniciando-se pelo token `[CLS]` e terminando-se em `[SEP]`, os tokens que mencionam uma determinada entidade são delimitados. Para isso, utilizaram o modelo pré-treinado $\text{BERT}_{\text{LARGE}}$ e, como córpus de treinamento, utilizou-se a Wikipedia em inglês, com blocos de parágrafos interligados.

Soares et al. (2019) realizaram experimentos em duas bases de classificação de relações: SemEval 2010 (HENDRICKX et al., 2019) e FewRel (HAN et al., 2018). A primeira contém 1.200 sentenças com relações anotadas, baseando-se em busca na Web. Entretanto, a segunda é derivada da *Wikipedia* e anotada por métodos de *crowdsourcing*, focada em *few-shot learning*.

O método MTB obteve $89,5\%$ de F1 contra $71,5\%$ do método de predição de relações TACRED (ZHANG et al., 2017) na base SemEval 2010. Além disso, o MTB também obteve $89,2$ *10-way 1-shot*[8] na base FewRel contra $94,3\%$ de humanos.

Por fim, vale ressaltar que existe uma implementação aberta desse trabalho. Além disso, entre as principais contribuições do método se destaca a extração de relação em texto não estruturado.

3.2 Abordagens no domínio do e-commerce

Esta seção apresenta métodos de construção e aprimoramentos de KGs no domínio do e-commerce. Na seção 3.2.1, apresenta-se o uso de grafos de comportamento para aprimorar um KG. Em seguida, a seção 3.2.2 mostra como lidar com termos subjetivos presentes na busca de produtos. A seção 3.2.3 descreve a construção automática de um KG no domínio do e-commerce evidenciando as particularidades do domínio. Por fim, a seção 3.2.4 mostra como gerar regras explicáveis na predição de *links* utilizando um mecanismo de atenção.

3.2.1 *Bayes Embedding* (BEM) (YE et al., 2019)

O BEM propõe refinar *embeddings* pré-treinadas de Grafos de Conhecimento (KGs) e Grafos de Comportamento (BGs) [9]. Para tanto, o conhecimento de KG e BG é agregado

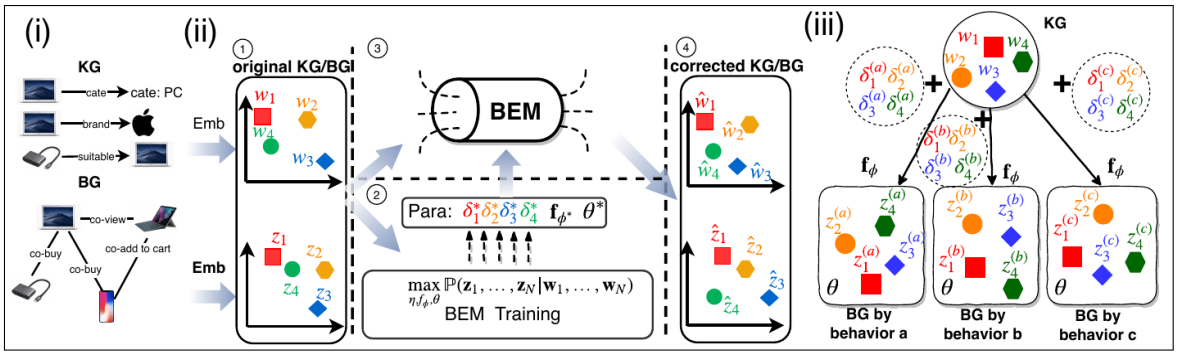
[8] Trata-se de um treino que contém 1 instância de uma única classe entre 10.

[9] Segundo Ye et al. (2019), um BG cobre uma grande variedade de grafos e redes convencionais, como a rede de links em páginas da internet (que modela o comportamento dos links em páginas da web),

por um *framework* Bayesiano. Desse modo, os autores afirmam que o BEM é capaz de refinar mutuamente as *embeddings* de ambos os grafos, ao mesmo tempo em que preserva suas estruturas topológicas.

Segundo os autores, é como se o KG fosse um conhecimento *a priori* e o BG fosse equivalente à experiência, usado como um fator de correção baseado no contexto. Por exemplo, como ilustrado na Figura 8, em (i) é possível perceber que características do produto, como categoria (`cate`) e marca (`brand`) são relações encontradas em um KG, enquanto que interações do usuário como visualização (`co-view`) e compra conjunta (`co-add to cart` e `co-buy`) aparecem como relações em um BGs. Então, em (ii) o *framework* gera fatores de correção baseado nas *embeddings* do KG, os quais são aplicados nas *embeddings* do BG em (iii).

Figura 8 – Funcionamento do *framework* do BEM.



Fonte: (YE et al., 2019)

Para Ye et al. (2019), a integração entre BG e KG no domínio do e-commerce traz diversas vantagens. Segundo eles, por exemplo, se o BG detectar uma visualização conjunta de um vestido longo e sapatos de salto alto o KG permitiria recomendar acessórios corretos para esses itens, já que o KG é capaz de armazenar a informação de que eles são itens de vestuário formal. Além disso, BGs assistidos por BGs também conseguem embasar comportamentos em fatos, como a recomendação de casacos de inverno quando o usuário compra passagens para o Alaska.

Por outro lado, BGs podem ser usados para derivar novo conhecimento (novas entidades, por exemplo) com base na interação do usuário, como novas tendências de moda a partir de produtos que foram comprados e clicados de forma conjunta, com frequência, no último mês, por exemplo. Além disso, um nó do BG pode ser pensado como um fator de ajuste para a entidade correspondente no KG, como ilustra a Figura 8 partes (ii) e (iii). Por exemplo um celular, que conceitualmente é um eletrônico portável (KG), pode estar em cenários diferentes no BG, como uma ferramenta de comunicação, um objeto de entretenimento e uma ferramenta de estudo *online*.

redes de autor-citação (que modela o comportamento das citações) e o comportamento de interação item-item (que modela as relações de `co-click` ou `co-purchase`, por exemplo).

Segundo Ye et al. (2019), há duas diferenças fundamentais entre KG e BG. A primeira está na maneira como os grafos registram informações e na heterogeneidade dessas informações. No KG, o conhecimento é guardado em triplas (h, r, t) onde h, t são entidades e r a relação entre elas, enquanto que o BG registra conhecimento a partir de relações de comportamento entre dois produtos, gerando um grafo não direcionado. Além disso, os nós de um KG tendem a ser mais heterogêneos do que os nós de um BG. A segunda, está na persistência do conhecimento. No KG, triplas significam aspectos intrínsecos dos produtos, que persistem e não mudam ao longo do tempo, ao passo que no BG as relações são sensíveis a fatores externos do mundo real. Por exemplo, no verão é comum de se comprar óculos de sol e óculos de natação em conjunto, mas, no inverno, essa compra conjunta é provavelmente mais difícil de ocorrer. A diferença, portanto, entre ambos os grafos sugere que eles se complementam (YE et al., 2019).

Para os autores, o KG contém representações abstratas de uma entidade, enquanto o BG traz sua realização em algum contexto. Com base nessa ideia, eles alegam que é possível ver um BG como uma mistura de KG e algum fator de contexto específico (usado como um termo de ajuste), mas que normalmente só reflete algum aspecto do KG (como uma projeção nessa mistura). Essas suposições motivaram os autores a conectar KG e BG por meio de um modelo generativo. Assim, o objetivo do BEM, na prática, é atualizar *embeddings* de KG (z) maximizando a probabilidade

$$\sum_{(i,j):i\neq j} log(P(g(z_i, z_j)|(w_i, w_j))) \tag{12}$$

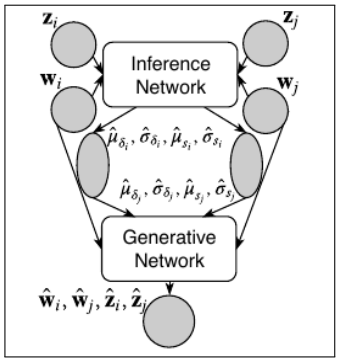
onde z_i, z_j são *embeddings* de KG, w_i, w_j são *embeddings* de BG e g é a função de arestas entre as *embedings* de KG z_i e z_j. Para isso, utiliza-se um *framework* bayesiano composto por dois modelos, como ilustra a Figura 9:

1. **Modelo de inferência** – responsável por extrair informações referentes à distribuição de probabilidade das *embeddings* de KG e de BG.

2. **Modelo generativo Bayesiano** – responsável por utilizar a informação extraída no modelo de inferência para atualizar a *embedding* de entidade no KG (w) com base na *embedding* equivalente no BG (z), criando um fator de correção δ. Para isso, foram criados três versões de modelos generativos: (1) o **BEM-P** que aplica a correção δ preservando as informações da interação entre cada par de entidades interligadas por uma relação; (2) o **BEM-I** que corrige analisando informações independentes sobre cada entidade, ou seja, desconsidera suas interações par-a-par; e (3) o **BEM-O**, que utiliza as *embeddings* originais, ou seja, não aplica a correção δ.

Na avaliação, Ye et al. (2019) realizaram experimentos para verificar se as *embeddings* refinadas pelo BEM se saiam melhores do que as pré-treinadas em algumas tarefas: clas-

Figura 9 – Estrutura de fluxo do BEM-P.



Fonte: adaptada de (YE et al., 2019)

sificação de nós, predição de *links*, classificação de triplas e recomendação de itens. Os autores também utilizaram as versões pré-treinadas das *embeddings* do TransE (BORDES et al., 2013b) e do TransD (JI et al., 2015b), obtidas com o uso do OpenKE[10]. Os testes foram realizados em três bases:

- **Duas bases pequenas** derivadas da base FB15K237[11] (DETTMERS et al., 2018) contendo 14.071 nós e 1.065.412 arestas. A primeira possui registros de *links* entre páginas da *Wikipedia*[12] (denominada FB15K237+*pagelink*); a segunda com registro de uma pequena descrição para cade entidade (denominada FB15K237+*desc*).

- **Base de larga escala** derivada de uma base de dados de produtos do *Alibaba Taobao*, contendo 17.37 milhões de entidades e mais de 5.000 relações.

Na tarefa de classificação de nós, as *embeddings* são usadas por um modelo de regressão logística multi-rótulo para treinamento e predição da classe em um conjunto de 46 rótulos possíveis. A Tabela 6 traz os resultados para essa tarefa, na qual as três versões do BEM foram avaliadas (BEM-P, BEM-I e BEM-O). Como pode ser visto pelos valores dessa tabela, a integração das *embeddings* de KG e BG proposta no BEM levou a melhores resultados em praticamente todas as avaliações, com destaque para o BEM-P na versão `concat`, que é a concatenação de ambas as *embeddings*.

Para CT e PL, devido ao fato de que o BEM refina apenas *embeddings*, o uso de *embeddings* pré-treinadas pelo TransE não se beneficia com a implementação do BEM; diferentemente do uso de *embeddings* do TransD, com as quais se observou uma breve melhora[13]. A Tabela 7 exemplifica como o uso do BEM afeta a tarefa de PL. Na primeira coluna, tem-se conceitos (entidades) e nas seguintes, a predição de categorias para o determinado conceito. A segunda não utiliza o BEM mas, a terceira, sim. Assim, nota-

[10] <https://github.com/thunlp/OpenKE>
[11] Obtida da base FB15K, excluindo-se as relações simétricas.
[12] <https://www.wikipedia.org>
[13] Para mais detalhes sobre os resultados obtidos, consulte (YE et al., 2019).

Tabela 6 – Resultado da classificação de nós utilizando acurácia (%).

		FB15K237 + *pagelink*					
		node2vec			LINE		
	BEM	KG	BG	concat	KG	BG	concat
TransE	O	85.59	75.12	89.39	85.59	77.57	89.44
	I	85.51	82.56	85.97	86.35	85.44	87.05
	P	**88.89**	**86.32**	**90.29**	**88.21**	**86.27**	**90.01**
TransD	O	86.06	75.12	89.18	86.06	77.57	89.00
	I	83.73	78.86	84.16	86.58	85.10	86.69
	P	**88.60**	**85.39**	**89.90**	**88.70**	**85.30**	**89.73**
		FB15K237 + *desc*					
		doc2vec			sentence2vec		
	BEM	KG	BG	concat	KG	BG	concat
TransE	O	85.32	75.62	**87.92**	85.32	83.42	88.43
	I	86.19	81.50	86.41	87.61	85.18	88.07
	P	**87.68**	**81.52**	87.86	**88.05**	**85.82**	**88.57**
TransD	O	85.83	75.62	88.07	85.83	83.42	88.52
	I	86.75	81.44	86.85	87.96	84.97	88.07
	P	**87.34**	**82.24**	**88.15**	**88.36**	**86.12**	**88.86**

Fonte: (YE et al., 2019)

se que o BEM consegue predizer de forma coerente mais categorias para o determinado conceito.

Tabela 7 – Exemplos da utilização do método BEM para predição de *links*.

concept	**predicted categories using original KG embedding**	**predicted categories using the KG embedding refined by BEM-P**
neuter clothing	jacket, homewear	Quick-drying T-shirt, sport down jacket, toning pants, aerobics clothes, warm pants
sports training	None	Quick-drying T-shirt, sports down jacket, sports bottle, Yoga T-shirt, training shoes, aerobics clothes
household items	succulents, detergent, tissue box, health tea, kitchen knife, man's facial cleanser, scented candle, washing cup, yoga mat towel	washing machine cover, spray, fish tank cleaning equipment, pen container, digital piano, pillow interior, wood sofa, table, bath bucket, composite bed, tape, mosquito patch, storage rack, needle, maker, storage box , leather sofa, indoors shoes, cotton swab, laundry ball, coffee cup, desiccant, trash bag

Fonte: Adaptada de (YE et al., 2019)

Além disso, na tarefa de recomendação de itens, avaliada na base de dados de larga escala, considerando a eficiência computacional, os autores usaram o TransE para gerar as *embeddings* do KG em uma base de conhecimento do Alibaba Taobao. Para as *embeddings* do BG, eles usaram o GraphSAGE[14] em um grafo construído a partir dos comportamentos

[14] Segundo os autores, o GraphSAGE é um exemplo de *Graph Neural Network*s (GNNs) que tem atingido boas performances em bases de dados de larga escala.

dos usuários, por exemplo, dois itens foram conectados se um certo número de clientes compraram eles simultaneamente nos últimos meses. A avaliação levou em consideração o número de itens recuperados que foram efetivamente comprados/clicados pelo usuário nos dias seguintes. A Tabela 8 traz os resultados dessa avaliação.

Tabela 8 – Resultado da tarefa de recomendação para as interações de comprados ou clicados pelo cliente.

Granularity	Hit @	click		buy	
		BEM-O	*BEM*-P	*BEM*-O	*BEM*-P
brand	10	15.97	**16.14**	24.87	**25.10**
	30	16.65	**17.12**	25.70	**26.57**
	50	17.26	**17.90**	26.39	**27.33**
category	10	**27.46**	27.40	27.85	**27.91**
	30	28.43	**29.99**	28.50	**29.45**
	50	29.58	**32.88**	29.26	**31.47**

Fonte: (YE et al., 2019)

Os autores checaram se os itens recuperados eram da mesma marca (*brand*) ou categoria (*category*) daqueles efetivamente comprados/clicados nos dias seguintes. Combinando essas duas granularidades, eles observaram que a taxa de acerto (*hit@k*, $k = 10, 30$ ou 50) do BEM-P foram $1 - 3\%$ melhores do que o BEM-O, o que eles relatam como bem significante considerando que haviam 9 milhões de itens. Segudo os autores, esses resultados validam que o BEM-P é capaz de incorporar informação útil do KG nas *embeddings* do BG para o propósito de recomendação de item.

Segundo os autores, o *framework* do BEM é geral e flexível no sentido de que ele pode receber como entrada quaisquer *embeddings* pré-treinadas de KG e BG, e refiná-las mutualmente. Como trabalhos futuros eles mencionam a alteração no código para permitir o treinamento conjunto das *embeddings*, o que poderia trazer ainda mais ganhos para as tarefas investigadas.

Por fim, vale mencionar que a implementação do BEM está disponível abertamente em: <https://github.com/Elric2718/Bayes_Embedding>

3.2.2 *Emerging Query Terms* (JIANG et al., 2019)

Jiang et al. (2019) propõem um método de completude de KG[15] de produto utilizando termos emergentes em *queries* (*emerging query terms*). Cada tripla nesse KG é da forma (`Categoria-Propriedade-Valor`), onde: `Categoria` é uma coleção de itens de produtos; `Propriedade` é cada uma das características (predicados) de uma categoria; e `Valor` é cada um dos valores que uma propriedade pode assumir.

[15] Jiang et al. (2019) explicitam que utilizaram uma Base de Conhecimento (KB) composta por triplas *Category-Property-Value* (CPV) que podem ser vistas como triplas SPO. Por isso, para esta seção, considera-se KB como um KG.

A ideia é classificar os termos emergentes, identificados nas *queries*, nas propriedades correspondentes nas categorias da taxonomia existente. Os autores consideram como termos emergentes os que são subjetivos e que ainda não estão no KG. Por exemplo, na busca de produtos que contenham a "imagem da Hello Kitty" a provável intenção do usuário é encontrar diversos itens com a imagem da Hello Kitty, e não uma imagem em si.

Segundo Jiang et al. (2019), KGs de domínio específico sempre estão incompletos, o que prejudica as aplicações que os utilizam. Isso se dá devido à construção manual que não leva em consideração as buscas subjetivas e os valores relevantes para a experiência do usuário. Por exemplo, se o KG foi construído por *experts* do domínio para ser capaz de identificar roupas com estampa de "animais", o usuário pode ter dificuldade de achar produtos relevantes se quiser uma estampa de "cachorro". A esses fatos somam-se a baixa cobertura (e alta precisão) da geração manual dessas bases de conhecimentos, feitas por *experts*; e o surgimento constante de novos termos, característica inerente ao domínio do e-commerce.

Para tentar contornar essa incompletude, os autores propõem utilizar o registro de *queries*. Primeiramente, porque os termos de uma *query* são emergentes, ou seja, refletem a tendência de estilo de busca em um determinado momento e, como tal, ajudam a manter o KG sempre atualizado. Além disso, ao construir um KG a partir de *queries*, o conhecimento é aprendido com base na experiência do usuário; o que é fundamental para a construção de um KG que entende "melhor" o consumidor (JIANG et al., 2019).

Assim, a proposta é avaliar a relevância entre um termo emergente e valores nas propriedades no KG. Os autores ressaltam que essa complementação do KG é essencial para mantê-lo completo e atualizado, e que, na proposta atual, apenas novos valores de propriedades são descobertos e não novas categorias ou propriedades.

Para isso, utilizam-se dois mecanismos de exploração de grafos: random walks (RWs) e *shortest path* (SP) encontrando benefícios em evidências (triplas) positivas e malefícios em evidências negativas. As evidências positivas são calculadas com base na premissa de que "quanto mais similares forem dois termos, maior a chance deles pertencerem à mesma propriedade". Vários fatores são usados para calcular as evidências positivas, como a similaridade entre os termos, calculada via distância de edição (Levenshtein) e similaridade de cosseno em *word embeddings* (word2vec) treinadas com os títulos dos itens clicados. Para as evidências negativas, considerou-se, por exemplo, a co-ocorrência de termos nas *queries*, uma vez que termos que co-ocorrem com frequência tendem a pertencer a propriedades diferentes; a distribuição dos termos nas categorias; a similaridade de *part-of-speech*, entre outros.

Assim, o objetivo é classificar cada termo emergente de uma *query* em uma propriedade correspondente, quantificando *scores* de termos com base no quão provável seria o termo pertencer à propriedade, como indica a Equação 13:

$$propriedade(t) = arg_{p \in P} max[f_{score}(p, t)] \tag{13}$$

onde P é um conjunto de propriedades, t é o termo emergente e f_{score} é a função de *score*. Para cada termo (tanto emergente quanto valores de propriedade), coletam-se as evidências positiva e negativa para entender se dois termos pertencem à mesma propriedade. Em seguida, constrói-se um grafo de similaridade de termos[16] como o ilustrado na Figura 10, no qual cada aresta representa a conexão entre termos que pertencem à mesma propriedade e os nós `chinese red` e `cake sleeves` representam termos emergentes. Por fim, utiliza-se o mecanismo de exploração de grafos (RW e SP), excluindo caminhos encontrados na evidência negativa, com o intuito de calcular a probabilidade de um termo emergente pertencer à uma propriedade. Dessa forma, o Algoritmo 2 sumariza a solução proposta por Jiang et al. (2019).

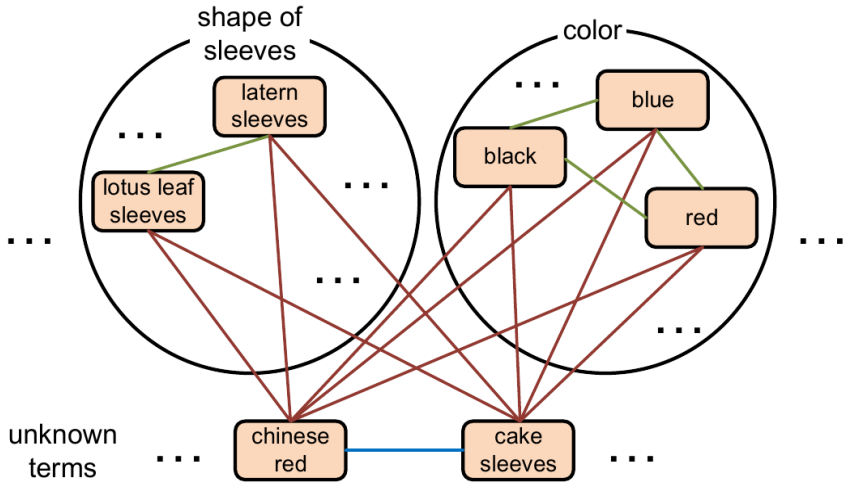
Algoritmo 2 Termos Emergentes em *queries*

Require: Entidade h; Propriedades de h P; Valores V para cada p; Termos U; Queries Q;

1: $E_{positiva}, E_{negativa} =$ **colete_evidência**$(U \cup V)$
2: $G \leftarrow$ **gere_grafo_similaridade**$(E_{positiva})$
3: **explore**$(grafo = G, restrições = E_{negativa})$
4: $\hat{p}(t) = arg_{p \in P} max[f_{score}(p, t)]$ **para cada** $t \in U$
5: **return** $\hat{p}$

Fonte: Adaptado de (JIANG et al., 2019);

Figura 10 – Exemplo de grafo de similaridade.



Fonte: (JIANG et al., 2019)

É interessante notar também que categorias diferentes podem associar significados diferentes para um dado valor de propriedade. Por exemplo, o termo "grande" na categoria

[16] Para mais detalhes sobre o cálculo das similaridades, consulte (JIANG et al., 2019).

de ar condicionado é diferente do termo "grande" na categoria de vestidos. Por isso, aplicou-se a solução proposta para cada categoria.

O método foi avaliado utilizando SP ou RW – sob restrições de evidências negativas – na tarefa de complementação de KG. Os modelos foram comparados a alguns métodos, entre eles: LDA (BLEI; NG; JORDAN, 2003), DF-LDA (ANDRZEJEWSKI; ZHU; CRAVEN, 2009) e METIC (XU et al., 2018); e também foi comparado com o método TransE (BORDES et al., 2013b) com uma Rede Neural Convolucional (CNN), referência em completude de KGs. Os experimentos ocorreram com uma base formada por propriedades de produtos[17] e filmes[18], contendo, no total: 5 categorias, 3.079 termos, 2.512 valores para as propriedades e 567 termos emergentes.

Os resultados mostraram que os modelos propostos possuem potencial. Por exemplo, o modelo utilizando *random walk* obteve *hits@5* de $0,96$ na categoria *Air conditioner* contra $0,91$ do melhor *baseline*. As Tabelas 9 e 10 ilustram os resultados obtidos em *hits@k*.

Tabela 9 – Resultados dos métodos avaliados nas categorias *Dress, Air conditioner, Perfume* e *T-shirt* em *hits@k*.

Hits@k	Dress			Air conditioner			Perfume			T-shirt		
	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5
Max-ES	0.47	0.61	0.72	0.34	0.63	0.81	0.47	0.75	0.90	0.36	0.56	0.69
Avg-ES	0.32	0.51	0.61	0.17	0.35	0.76	0.31	0.64	0.85	0.28	0.50	0.66
PE	0.51	**0.88**	**0.93**	0.39	0.77	0.91	0.50	0.78	0.93	0.40	0.79	**0.89**
LDA	0.25	0.43	0.65	0.24	0.56	0.66	0.32	0.79	0.90	0.14	0.42	0.63
DF-LDA	0.24	0.39	0.49	0.20	0.50	0.61	0.31	0.73	0.88	0.10	0.33	0.58
METIC	0.07	0.32	0.37	0.56	0.72	0.72	0.55	0.72	0.76	0.19	0.28	0.37
NRE	0.33	0.39	0.46	0.31	0.33	0.42	**0.60**	0.75	0.80	0.15	0.23	0.40
transE+CNN	0.11	0.27	0.42	0.15	0.40	0.63	0.12	0.40	0.72	0.11	0.27	0.43
SP	**0.59**	0.79	0.85	**0.58**	0.77	0.93	0.48	**0.82**	**0.94**	**0.44**	0.74	0.86
RW	0.51	**0.88**	**0.93**	0.40	**0.79**	**0.96**	0.50	0.81	**0.94**	0.40	**0.80**	**0.89**

Fonte: (JIANG et al., 2019)

Os autores também apresentam resultados da avaliação das evidências positivas e negativas adotadas na proposta e concluem que, ambas, são efetivas e trazem impacto significativo. Os autores também apontam o desempenho do método proposto em relação aos métodos neurais METIC, NRE e transE+CNN alegando que a pouca quantidade de dados nesse domínio específico limitou a performance desses métodos.

Jiang et al. (2019) também relatam que o modelo treinado com RW para o KG do CPV foi colocado em produção e que as top-5 propriedades mais relevantes para cada termo emergente foram avaliadas manualmente. Após esse processo, eles relatam que a porcentagem de *queries* não reconhecidas caiu de 69,10% para 43,57%, e a de termos não reconhecidos caiu de 69,19% para 36,31%. Eles acrescentam, ainda, que quanto

[17] Especificada como CPV KB no artigo.
[18] Derivada da base CN-DBpedia (XU et al., 2017).

Tabela 10 – Resultados dos métodos avaliados na categoria *Movies* em *hits@k*.

Method	Hits@1	Hits@3	Hits@5
Max-ES	0.19	0.47	0.64
Avg-ES	0.00	0.00	0.01
PE	0.24	0.55	0.72
LDA	0.01	0.07	0.28
DF-LDA	0.06	0.07	0.28
METIC	0.08	0.50	0.64
NRE	0.12	0.52	0.67
transE+CNN	0.02	0.05	0.08
SP	**0.25**	0.55	0.72
RW	0.24	**0.59**	**0.76**

Fonte: (JIANG et al., 2019)

mais completo for o KG, melhor será a precisão do método, uma vez que mais preciso provavelmente será o cálculo da função de *score*.

Por fim, os autores não disponibilizaram o código abertamente.

3.2.3 *Product Knowledge Graph* (XU et al., 2020)

Xu et al. (2020) propõem o um método para geração de Grafo de produtos (PKG), elaborado pelo Grupo "Walmart Labs". Assim, o método usa uma base de dados composta por: descrições de produtos (textos pequenos contendo de 20 a 100 palavras), histórico de navegação (busca, substituição de produtos) do usuário e a hierarquia (categorias, subcategorias, departamentos e super-departamentos). O modelo se baseia nas relações `search` e `describe` para sumarizar as interações entre língua natural (presente nas descrições dos produtos) e produto (presente nas atividades de busca). Além disso, tratando produtos, palavras e rótulos de categorias como entidades e relações como arestas, o conhecimento multi-relacional envolvendo os produtos pode ser sumarizado em um PKG como o ilustrado na Figura 11.

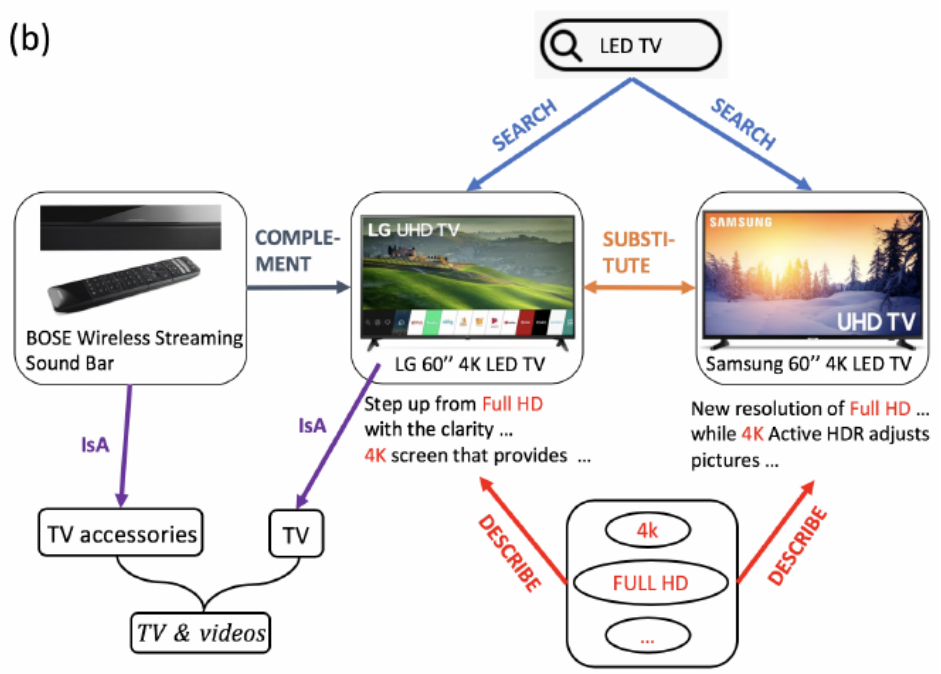
Xu et al. (2020) trazem uma comparação entre PKGs e KGs genéricos em vários aspectos, dos quais destacamos:

1. **Dados** – KGs genéricos representam fatos na forma de triplas SPO, ou (h, r, t). Já os dados em um PKG possuem múltiplas modalidades, incluindo informações do catálogo do produto (por exemplo, descrições e categorias) e registros de interações entre usuário e produto.

2. **Modelo** – em PKGs, observações de fatos são mais suscetíveis a ruído.

3. **Relações semânticas** – relações entre produtos do PKG possuem valor semântico mais significativo, uma vez que os produtos possuem diversos propósitos no mundo real.

Figura 11 – Esquema de um grafo de produtos.



Fonte: Adaptado de (XU et al., 2020)

4. **Informações adicionais** – ambos utilizam descrições e dados textuais como informações adicionais.

5. **Regras Lógicas** – enquanto é comum utilizar cláusulas de Horn[19] em KGs, relações em um PKG podem desobedecê-las. Por outro lado, é possível propagar regras de similaridade entre entidades ou relações de produtos substituíveis entre si.

Diante disso, o modelo propõe a extração de 6 relações: `substitute`, `complement`, `co-view`, `search`, `describe` e `isA`. Abordagens diferentes foram propostas para lidar com cada relação:

1. **Relação `substitute`** – os autores partem da suposição de que produtos parecidos ou substituíveis provavelmente possuem propriedades similares; o que denominam de "regra da propagação" (*propagation rule*). Com base na hipótese distribucional – que diz que palavras contextualmente similares têm representações similares – a regra da propagação implica que as *embeddings* de produtos substituíveis e_1 e $e_2 \in E$ estejam geometricamente próximas em um espaço de *embedding* de produto.

2. **Relações `complement`, `co-view`, `describe` e `search`** – os autores adaptaram uma camada de *self-attention* para cada relação com o intuito de recuperar as relações-alvo a partir de informações ruidosas vindas das descrições de produto e dos registros

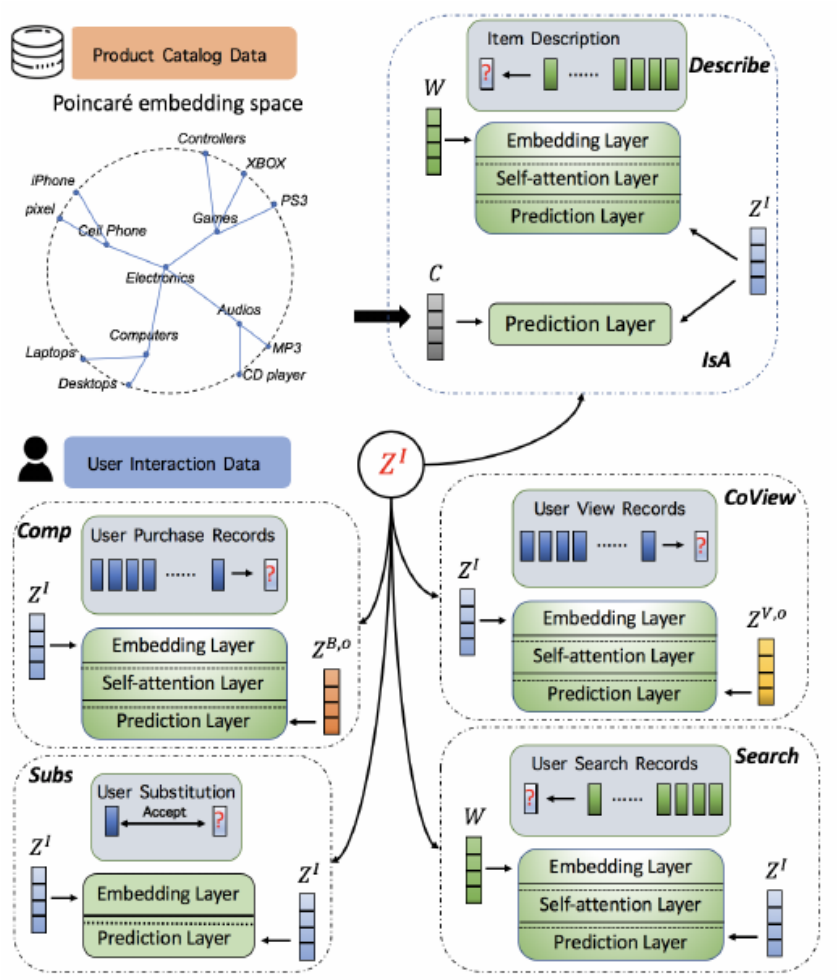
[19] Como regra, uma cláusula de Horn pode ser definida como uma conjunção de fatos $A \Leftrightarrow (a_1 \wedge a_2 \wedge ... \wedge a_n)$ que levam em outro fato B, ou seja, se todos os a_i em A ocorrem, então B também ocorre $((a_1 \wedge a_2 \wedge ... \wedge a_n) \rightarrow B)$. No contexto de KGs, uma regra indica que a ocorrência conjunta de algumas triplas pode significar que uma nova tripla pode ser inferida.

de atividades dos clientes. A ideia aqui está alinhada com o mecanismo de atenção proposto na tradução neural (VASWANI et al., 2017), no qual uma sentença é representada por uma soma ponderada de representações individuais de palavras.

3. **Relação `isA` e hierarquia de categorias** – os autores lidam com essas relações utilizando *embeddings* no espaço *Poincaré ball* (*Poincaré Embeddings* (NICKEL; KIELA, 2017)), por se tratarem de relações baseadas em uma estrutura hierárquica em árvore.

A Figura 12 traz a arquitetura do modelo, treinado seguindo a abordagem de (SANH; WOLF; RUDER, 2019).

Figura 12 – Arquitetura proposta para o treinamento do modelo KG.



Fonte: (XU et al., 2020)

Para avaliar o modelo, os autores utilizaram uma base de dados[20] contendo cerca de 140.000 produtos, suas descrições (contendo de 20 a 100 palavras) e sua hierarquia de

[20] Obtida a partir de <grocery.walmart.com>

categorias (contendo 9 super-departamentos, 28 departamentos, 228 categorias e 1.198 subcategorias). Além disso, utilizou-se cerca de 40 milhões de registros de sessão de usuário, contendo visualizações, compras, *queries* de busca e histórico de cliques. Por fim, utilizou-se, também cerca de 70 mil registros de produtos substituídos.

Entre os experimentos realizados, avaliou-se o desempenho do método proposto nas tarefas de predição de links, recomendação e ranqueamento de busca:[21]

- **Predição de links** – o modelo de (XU et al., 2020) foi o melhor na predição de links para as relações `complement`, `co-view` e `substitute`, obtendo, respectivamente, valores de *hits@10* (seção 2.3.1.1) de $14,53$, $20,84$, e $34,58$ contra $7,81$, $12,38$ e $31,25$ do melhor modelo *baseline* em comparação, que foi o ComplEx (TROUILLON et al., 2016).

- **Recomendação** – o modelo de (XU et al., 2020) também foi o melhor nesta tarefa com *hits@10* (seção 2.3.1.1) de $13,72$ contra $11,30$ do modelo *triple2vec* (WAN et al., 2018).

- **Ranqueamento de busca** – o modelo de (XU et al., 2020) também foi o melhor nesta tarefa com uma medida de cobertura *top-10* (Recall@k – R@k[22]) de $30,99$ contra $21,58$ do modelo ComplEx (TROUILLON et al., 2016).

Os autores apontam que os resultados empíricos obtidos em uma base de dados real mostram que o modelo proposto é promissor, superando, principalmente, a performance do *baseline ComplEx* nesse sentido. Além disso, concluem que, além das tarefas básicas em KGE, o modelo também se beneficia nas tarefas de busca e recomendação.

O artigo não disponibiliza o código abertamente, não explicita a base de treinamento e também não disponibiliza a base de teste. Contudo, é interessante destacar o uso de *self-attention* do trabalho de Xu et al. (2020).

3.2.4 XTransE (ZHANG et al., 2020)

Zhang et al. (2020) apresentam um método para a construção de um grafo de conhecimento, no domínio do e-commerce, que é baseado no TransE: o XTransE. Esse método modela produtos, suas propriedades e a pertinência de um produto a um *Lifestyle* (relação `belongTo`); além de explicar resultados através de regras compreensíveis para seres humanos. Esse *Lifestyle* está relacionado ao contexto no qual o produto está inserido. Por exemplo, um *Lifestyle* sugere um bonsai para quem procura um vaso de plantas na plataforma e-commerce.

[21] Para detalhes de todas as avaliações do método, consulte (XU et al., 2020).

[22] A medida de cobertura *top-k* consiste em entender quantos dos k exemplos melhor ranqueados são relevantes em relação ao total de exemplos relevantes no conjunto de teste, ou seja: $cobertura@k = \frac{\#relevantesentreoskmelhorranqueados}{\#relevantes}$.

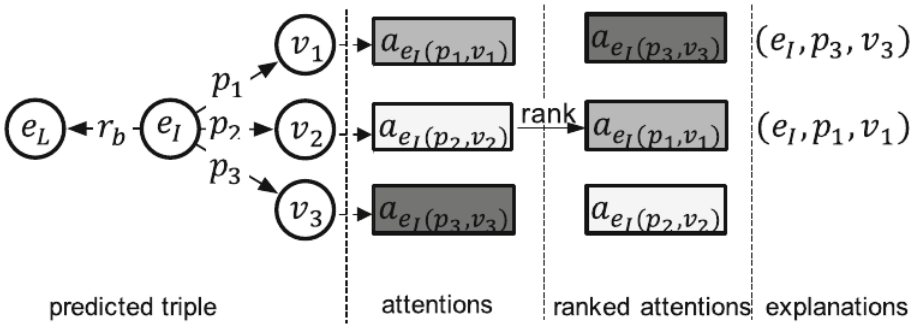
Os autores, então, formulam o problema como o de predição de *links* onde as triplas representam propriedades dos itens, como (`iPhoneXS-01`,`color`,`Black`) e a relação `belongTo` representa a ligação entre itens e *lifestyles*.

Assim, os autores propõem um método para construção de um KGE, um gerador de explicações para resultados preditos e um gerador de regras.

1. **Construção do KGE** – os autores utilizaram o método TransE (BORDES et al., 2013b) com um mecanismo de atenção. Assim, é possível mensurar quanto uma relação item-propriedade influencia a relação item-*lifestyle*. Portanto, trata-se de um modelo de *embedding* capaz de explicar predições de arestas item-*lifestyle* (daí o nome XTransE que deriva de "*Explainable TransE*").

2. **Geração de explicações** – as explicações são geradas com base na atenção calculada nas relações item-propriedade. Portanto, para uma tripla item-*lifestyle* predita (i, r_b, l) onde i é um item, l é um *lifestyle* e r_b é a relação `belongsTo`, o gerador ranqueia todas as atenções $a_{i(r_p,v)}$ (onde i é um item, r_p é uma relação de propriedade e v o valor dessa propriedade) e seleciona as triplas item-propriedade entre os k primeiros valores de atenção como explicações. Na Figura 13, entende-se que a tripla (e_I, r_b, e_L) é gerada pelas triplas (e_I, p_1, v_1) e (e_I, p_3, v_3), já que essas obtiveram valores de atenção suficientemente maiores (quanto mais escura a cor na figura, maior o valor associado).

Figura 13 – Exemplo de geração de explicações.



Fonte: (ZHANG et al., 2020)

3. **Geração de regras** – com as explicações geradas, é possível entender que uma tripla item-*lifestyle* (i, r_b, l) é explicada pelas triplas $(i, r_{p_1}, v_1), (i, r_{p_2}, v_2), ...(i, r_{p_n}, v_n)$. Considerando, também, que uma regra consiste em uma cabeça h e um corpo b tais que $h \leftarrow b$ ou $h \leftarrow b_1 \wedge ... \wedge b_n$, o gerador de regras entende que a cabeça de uma regra é uma tripla item-*lifestyle* e o corpo é composto pela composição de um conjunto de triplas item-propriedade que explicam a primeira; como ilustram os exemplos da Figura 14. Uma tripla explicada pode ter múltiplas regras. Por exemplo, uma tripla com três explicações gera $C_3^1 + C_3^2 + C_3^3 = 7$ regras.

Figura 14 – Extrações de diferentes regras, com seus respectivos corpos e cabeças.

Rule head	Rule body
(*item X*, scene, making dessert)	(*item X*, category, cookware)
(*item X*, scene, making music)	(*item X*, category, news music) ∧ (*item X*, hasProperty, ISBN)
(*item X*, scene, boxing)	(*item X*, hasProperty, size) ∧ (*item X*, titleContains, protective clothing)
(*item X*, scene, sketch)	(*item X*, hasProperty, brand) ∧ (*item X*, hasProperty, suitable age) ∧ (*item X*, category, painting supplies)
(*item X*, scene, outdoor survival)	(*item X*, hasProperty, brand) ∧ (*item X*, hasProperty, color) ∧ (*item X*, hasProperty, tag price) ∧ (*item X*, hasProperty, origin)

Fonte: Adaptado de (ZHANG et al., 2020)

Para a avaliação do método, os autores utilizaram uma base de dados item-*lifestyle* composta por 72.849 entidades, 758 relações de propriedades, uma relação `belongTo` e 1.875.438 triplas. Com isso, para a tarefa de predição, comparou-se o modelo XTransE com o TransE (BORDES et al., 2013b) e o modelo proposto obteve MR (seção 2.3.1.2) de $1,10$ contra $1,86$ do TransE. Em termos de acurácia, XTransE obteve $91,42\%$ enquanto TransE alcançou $60,53\%$. Segundo os autores, isso mostra que um método de construção de KGE genérico precisa ser ajustado para tratar problemas de aplicações reais, principalmente quando o foco está em poucos tipos de triplas relacionais.

A avaliação das explicações foi realizada em uma amostra aleatória de triplas de teste contendo 500 corretamente preditas e 500 incorretamente preditas, as quais foram avaliadas manualmente para checar se as explicações eram consistentes com as predições. Os resultados mostraram que há mais explicações consistentes $(37,4\%)$ que inconsistentes $(11,6\%)$, tanto em corretas $(46,2\%$ contra $10,4\%)$ quanto nas incorretas $(28,6\%$ contra $12,8\%)$, o que indica a efetividade de utilizar o mecanismo de atenção para gerar explicações. Além disso, foram aprendidas 252 regras das explicações geradas[23].

Os autores concluem que com os experimentos em dados reais de e-commerce eles comprovam que o XTransE é melhor do que os *baselines* avaliados (em especial, o TransE), e que podem gerar explicações e regras úteis para a predição de relações item-*lifestyle*.

Por fim, vale mencionar que o XTransE está disponível em <https://github.com/wencolani/XTransE>.

3.3 Comparação entre os métodos

A Tabela 11 traz um resumo comparativo das principais características dos métodos descritos neste capítulo.

A contribuição desses trabalhos é notável, tanto no âmbito de representação e extração de conhecimento quanto no entendimento das particularidades do domínio do e-commerce.

[23] Para detalhes do aprendizado de regras, consulte (ZHANG et al., 2020).

Tabela 11 – Comparação entre os métodos.

Método (Seção)	**Principal contribuição**	**Tarefas finais**	**Resultados**	**Comparação com**
TransE (3.1.1)	- Método Simples e eficiente - Precursor de métodos de KGE baseados em Translação	Predição de *links*	$89,2\%$ hits@10 WN18	$52,8\%$ (RESCAL)
ComplEx (3.1.2)	- Tratamento de relações assimétricas - Refinamento dos métodos anteriores	Predição de *links*	$84,0\%$ hit@10 FB15k	$64,1\%$ (TransE)
BERT MTB (3.1.3)	- Extração de relação em texto não estruturado introduzindo *blanks*	Classificação de relações	- $89,2\%$ 10-*way* 1-*shot* FewRel - $89,5\%$ F1 SemEval 2010 8	- $85,88\%$ (humanos) - $71,5\%$ (TACRED)
BEM (3.2.1)	- Uso de um grafo de comportamento	- Classificação de nós - Predição de *links* - Classificação de triplas - Recomendação de itens	- $90,29\%$ acurácia, *embeddings* TransE - $43,66\%$ hits@10, *embeddings* TransE - $77,13\%$ acurácia, *embeddings* TransE - 31.47% hits@50, recomendações de compra por categoria	- $89,39\%$ (sem correção δ) - $43,14\%$ (sem correção δ) - $76,56\%$ (sem correção δ) - $29,26\%$ (sem correção δ)
EQT (3.2.2)	- Tratamento de termos subjetivos - Métodos simples de caminhada em grafos (SP e RW)	Predição de *links*	$76,0\%$ hits@5 categoria *Movies*	$8,0\%$ (TransE+CNN)
PKG (3.2.3)	- Evidencia particularidades do domínio do e-commerce - Tratamento particular para relações de interesse diferentes	- Predição de *links* - Recomendação - Ranqueamento de busca	- $34,58\%$ hits@10 `substitute` - $13,72\%$ hits@10 - $30,99\%$ R@10	- $31,25\%$ (ComplEx) - $11,30\%$ (*triple2vec*) - $21,58\%$ (ComplEx)
XTransE (3.2.4)	- Uso do mecanismo de atenção para gerar regras explicáveis à humanos	Predição de *links*	$1,10$ MR	$1,86$ MR (TransE)

Entretanto, devido a limitações de tempo e recursos, os experimentos desenvolvidos neste trabalho – detalhados no Capítulo 4 – utilizam apenas os métodos TransE e o ComplEx.

Capítulo 4

Geração de KG para o e-commerce

Com base no conteúdo apresentado anteriormente, este trabalho teve como objetivo a construção automática de KG no domínio do e-commerce. Assim, foi construída uma estrutura como a apresentada na Figura 15. Para a compreensão dessa figura, considere:

- **Item** – é a entidade que possui o conjunto de características de um determinado produto. Além disso, um item pode pertencer à mais de uma WIT. Por exemplo, o item `Cadeira Charles Eames Wood` pode pertencer às WITs `sala de estar` e `cadeira`.

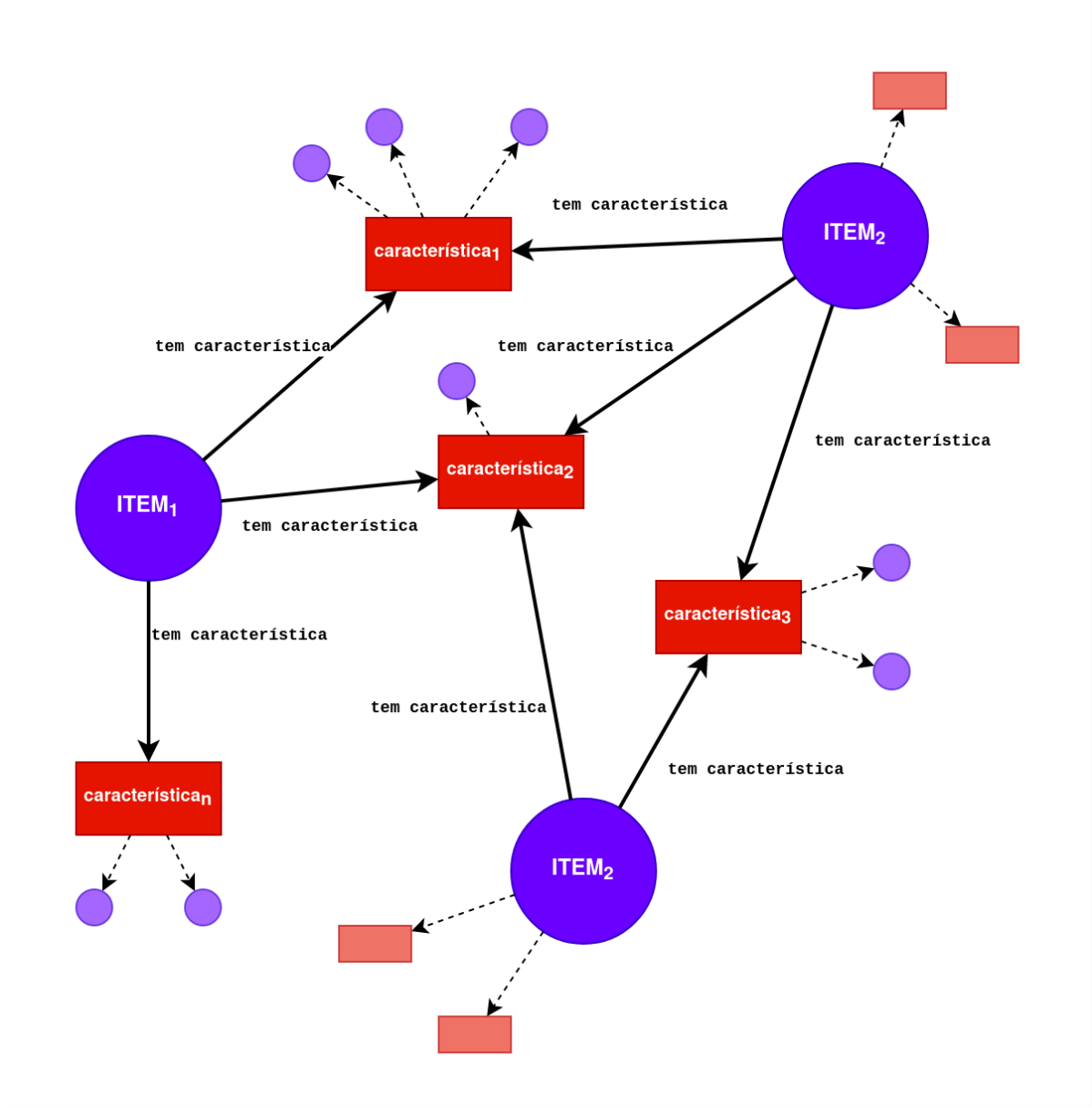
- **WIT** – é uma abreviação para *What Is This* e representa um agrupamento de itens com características suficientemente distintas de outras WITs. Por exemplo, `celular` é a WIT do item `Galaxy S10`, diferentemente de fone de ouvido, que é a WIT do item Redmi Airdots 3.

- **Atributo** – é o rótulo (nome) dado a uma característica de um item. Por exemplo, são atributos as propriedades modelo, cor, armazenamento e processador de um celular.

- **Valor** – é o dado numérico ou textual vinculado a um atributo. Por exemplo, o valor do atributo `cor` pode ser `vermelho` ou `branco` e o valor do atributo `armazenamento` pode ser `16gb` ou `32gb`.

Como ilustrado na Figura 15, o intuito é que o KG contenha informações suficientemente relevantes e úteis para as aplicações de interesse, como a especificação de quais características estão presentes em mais de um produto.

Figura 15 – Componentes no domínio do e-commerce dispostos em um esquema de KG.



4.1 Descrição do problema

Espera-se que os métodos desenvolvidos neste trabalho possam ser utilizados para recomendar produtos baseando-se em relações entre entidades do KG. Essas entidades podem ser tanto itens quanto atributos ou valores de atributos.

Neste trabalho, abordam-se recomendações baseadas apenas no conteúdo dos objetos, sem considerar dados produzidos por usuários das plataformas de e-commerce. Neste contexto, considera-se que o usuário observa um item – denominado de item observado – e lhe é recomendo outros itens – denominados de itens recomendados, os quais possuem propriedades parecidas com o primeiro. Essas propriedades podem ser tanto características estruturadas em comum, como atributos ou termos nas descrições, quanto propriedades latentes parecidas representadas por *embeddings*.

Essa ideia está ilustrada na Figura 16. Nela, o item observado é um *smartwatch* (bordas pretas) e os itens recomendados (bordas verdes) são um fone de ouvido e um *smartphone* por terem propriedades parecidas com o primeiro. No entanto, outros itens como um eletrodoméstico ou um brinquedo não são recomendados, uma vez que não compartilham propriedades parecidas com o item observado. Por isso, esses não são recomendados (bordas vermelhas).

Figura 16 – Diagrama mostrando a tarefa de recomendação considerada neste trabalho. Por conta de suas propriedades, o item observado (bordas pretas) recomenda dois itens (bordas verdes) e deixa de recomendar outros dois (bordas vermelhas).

Fontes: próprio autor e <https://www.americanas.com.br/>. Último acesso: 16/04/2022.

4.2 Materiais

Essa seção descreve os materiais usados neste trabalho: (4.2.1) o córpus utilizado no experimento e (4.2.2) a ferramenta que possibilita a implementação de um dos métodos de construção de KGs abordados neste trabalho.

4.2.1 Córpus Americanas S.A.

A geração do KG foi feita a partir do córpus Americanas S.A., o qual é um recorte da base de dados de toda a companhia. Vale mencionar que, por questões de confidenci-

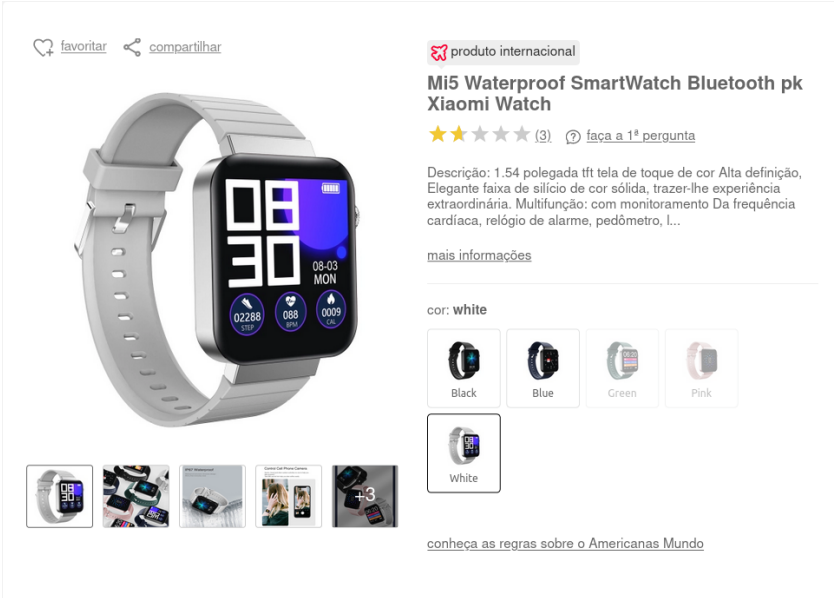
alidade, os dados brutos não serão apresentados neste documento. Assim, utilizou-se um recorte da base de dados referente aos produtos internacionais e divididos em dois grupos:

- **Conexão (BC)** – 459 produtos referentes ao nicho de produtos classificados como Mundo Conexão[1]. Cada exemplo foi escolhido manualmente por especialistas de domínio para compor este conjunto. No contexto da recomendação de produtos, esses são os produtos observados.

- **Sortimento (BS)** – 8.946 produtos candidatos à classe de Mundo Conexão. No contexto da recomendação de produtos, são os produtos candidatos a recomendação.

Esse córpus contém apenas informações não estruturadas de títulos e descrições dos produtos. A Figura 17 mostra um dos itens possíveis de se encontrar nesse nicho.

Figura 17 – Exemplo de item do Mundo Conexão. Nesta figura, é possível encontrar os dados não estruturados no canto superior direito.



Fonte: <https://www.americanas.com.br/>. Último acesso: 1º de abril de 2022.

Para definir as características de interesse na construção desse córpus, a companhia parceira levantou 5 WITs de interesse: `smartwatch`, `fone lenovo`, `caixa de som`, `mouse gamer` e `tablet`. A partir de cada WIT, também foram levantados sintagmas de interesse que podem ser encontrados nos dados não estruturados desses tipos de produtos e que caracterizam a classificação dos produtos como "Mundo Conexão". Esse levantamento foi um trabalho manual de analistas da empresa parceira. Esses estão listados na Tabela 12.

[1] <https://www.americanas.com.br/hotsite/amundo-21-conexao>

Tabela 12 – Sintagmas de interesse coletados para cada WIT.

WIT	**Sintagmas**
smartwatch	bluetooth, oxímetro, notificações de aplicativos do celular, touchscreen, relógio, bateria recarregável com carregamento USB, à prova d'água, alarme, conectividade com Alexa, monitor de frequência cardíaca, tempo meteorológico, pedômetro, cronômetro
fone lenovo	bluetooth, microfone, bateria recarregável com carregamento USB, à prova d'água, som HiFi, caixa de carregamento, bateria recarregável
caixa de som	bluetooth, à prova d'água, bateria de longa duração, auto-falantes potentes (10W), entrada pen drive, carregamento USB, entrada cartão de memória, reprodução de MP3, recarregável por USB, toca música, entrada para USB, entrada para Flash Drive, entrada para Cartões de Memória, adaptador bluetooth USB, conexão bluetooth, adaptador para áudio
mouse gamer	sensor ótico com precisão, botões programáveis, carregamento por USB, design ergonômico, Luzes LED, bateria recarregável, suporte ergonômico, conexão usb com fio
tablet	CPU, sistema operacional, camera, microfone, gps, tela touchscreen, entrada para chip celular, caixa de som, bluetooth, carregamento energia elétrica, memória ram, wifi, memória interna, bateria interna, bateria recarregável, conexão USB, memória medida em GB, conexão bluetooth, bateria

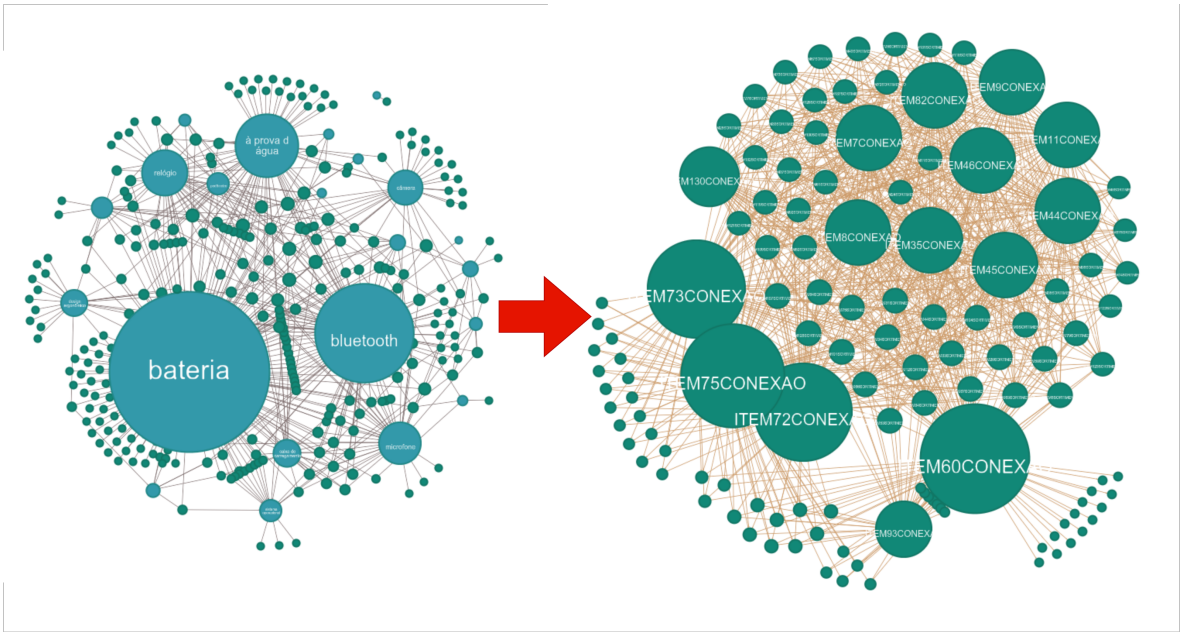
4.2.2 RedisGraph

Para servir de comparação com os métodos KGE, utilizou-se o banco de dados Redis[2] para armazenar as triplas. Especificamente, o Redis possui um módulo que opera um sistema de gerenciamento de banco de dados orientado a grafo, o RedisGraph[3] (CAILLIAU et al., 2019). Para consultas, o banco de dados faz uso da linguagem de *query* Cypher[4]. Essa é bastante semelhante à SQL, porém sua principal característica é a escrita simbólica das instâncias. A Figura 18 ilustra os grafos no RedisGraph produzidos neste trabalho através da plataforma de visualização RedisInsight[5].

Cailliau et al. (2019) relatam que o módulo processa *queries* rapidamente por convertê-las em operações de álgebra linear para grafos. Além disso, este trabalho interpreta a construção de KGs no RedisGraph como uma abordagem distributiva, uma vez que o módulo armazena seus elementos em uma matriz de adjacência comprimida.

[2] <https://redis.io/>
[3] <https://redis.com/modules/redis-graph/>
[4] <http://opencypher.org/>
[5] <https://redis.com/redis-enterprise/redis-insight/>

Figura 18 – KGs construídos utilizando o RedisGraph. À esquerda, está o grafo que liga itens a características; enquanto o grafo à direita liga itens diretamente a outros itens. Na seção 4.3.2.1, detalha-se como se parte do grafo à esquerda para construir o da direita.



4.3 Métodos

A Figura 19 ilustra as etapas do desenvolvimento deste trabalho. O módulo de extração de informações transforma o Córpus Americanas S.A. e as características de interesse em triplas de relações item-característica, as quais servem de entrada para o módulo de construção de KGs. Assim é possível criar um grafo através da plataforma RedisGraph e representações KGE dos elementos do grafo.

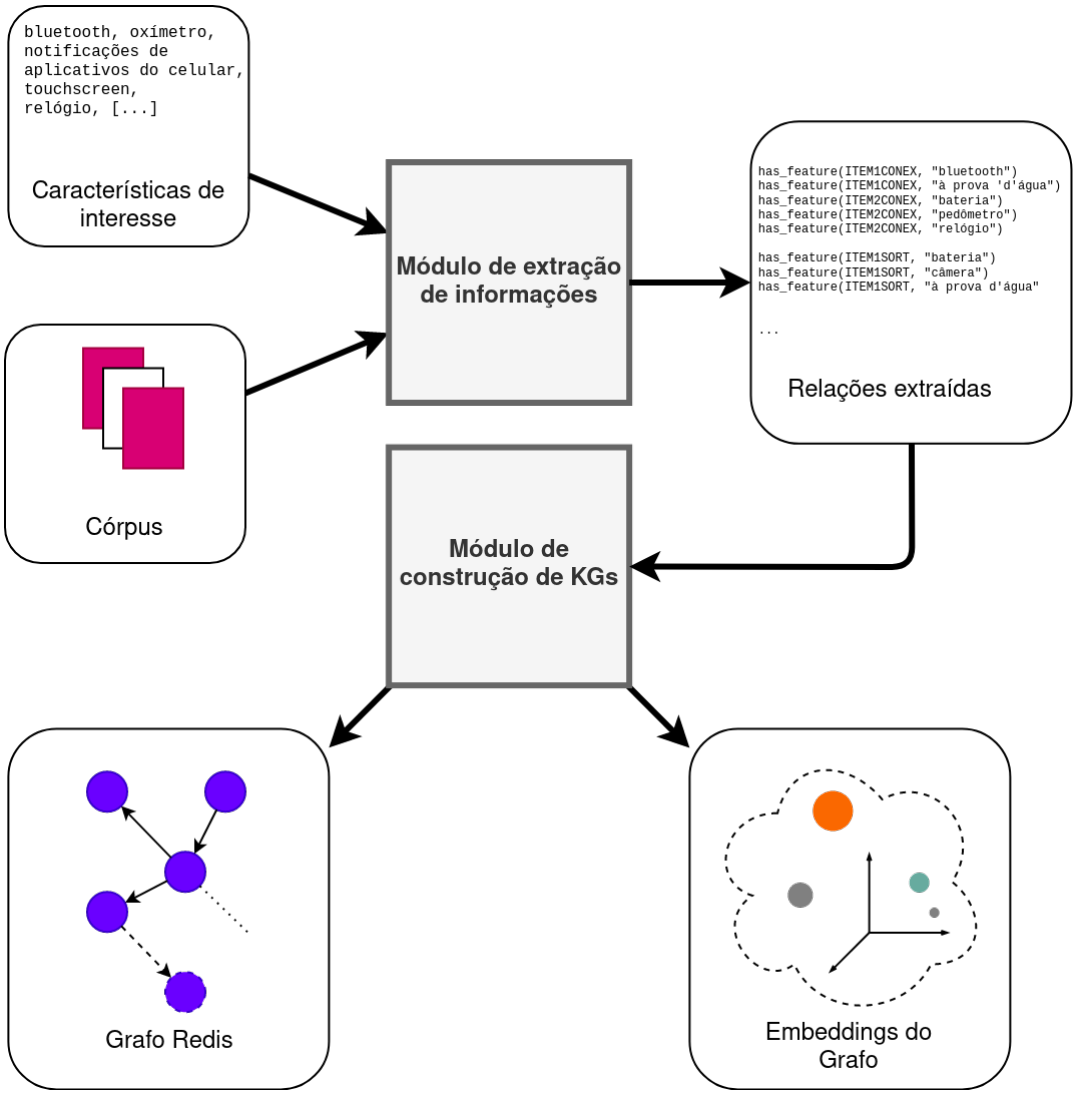
As subseções a seguir explicam as partes da solução desenvolvida neste trabalho: a seção 4.3.1 descreve o módulo de extração das informações do córpus; e a seção 4.3.2 especifica o módulo de criação de KGs a partir do que foi extraído pelo módulo em 4.3.1.

4.3.1 Módulo de extração de informações

Esta seção descreve o módulo cuja função é extrair informações do córpus descrito em 4.2.1. Seu objetivo é recuperar informação útil, ou seja, triplas SPO, em dados não estruturados de títulos e descrições de produtos, para construir um grafo de conhecimento. A Figura 20 ilustra essa extração.

Uma questão levantada no início do desenvolvimento deste trabalho foi a importância do uso das informações não-estruturadas no domínio do e-commerce. Para responder tal questão experimentos foram realizados, como descrito na subseção a seguir.

Figura 19 – Diagrama de funcionamento dos módulos e seus resultados.



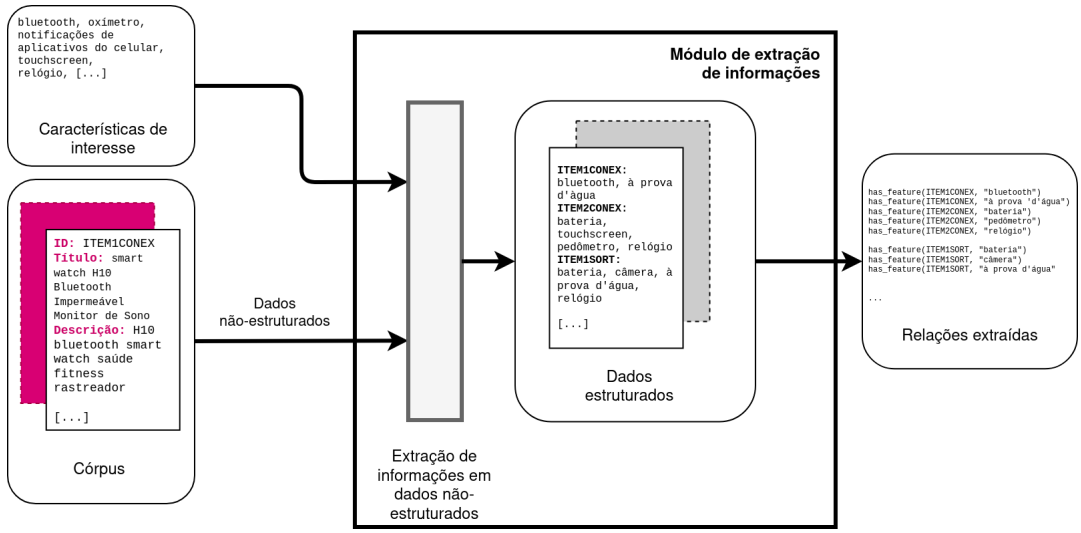
4.3.1.1 Explicando a relevância de extrair informações de dados não estruturados no domínio do e-commerce

Esta seção descreve os experimentos prévios (BARBIRATO; REAL; CASELI, 2021) realizados para extração de relações em dados não estruturados (título e descrição) em itens do tipo *smartphone* vendidos nas plataformas da companhia parceira. Como motivação para o desenvolvimento deste trabalho está a constatação de Xu et al. (2020) de que a informação no domínio do e-commerce é incompleta, e o conhecimento está latente nos dados não estruturados.

Assim, experimentos foram realizados para comparar os tipos e quantidades de informações derivadas de dados estruturados e não estruturados no domínio do e-commerce. Para isso, utilizou-se 956 itens do tipo *smartphones* e celulares de duas bases (B1 e B2):

B1 – 540 produtos, contendo títulos em português e 8 propriedades recuperadas de suas fichas técnicas.

Figura 20 – Diagrama de funcionamento do módulo de extração de informações. Munido das características de interesse, o módulo extrai informações de dados não estruturados, transformando-os em informações estruturadas para criar triplas.



B2 – 416 títulos de produtos e respectivas entidades nomeadas (SILVA et al., 2021). Trata-se de uma base em português, anotada por 2 linguistas com as seguintes entidades nomeadas: `Modelo`, `Marca`, `Cor`, `Memória Interna`, `Sistema Operacional`, `Capacidade de chips`, `Tamanho de tela`, `Câmera traseira` e `Processador`.

As Tabelas 13 e 14 trazem algumas propriedades da B1, em destaque aquelas que correspondem às entidades nomeadas anotadas na B2. A Tabela 21, por sua vez, traz um exemplo de título de produto anotado, no formato encontrado na B2.

Tabela 13 – Propriedades mais frequentes da ficha técnica em toda a base B1.

atributo	#	%
garantia do fornecedor	540	99,82 %
conteúdo da embalagem	540	99,82 %
alimentação tipo de bateria	539	99,63 %
banda	538	99,45 %
filmadora	538	99,45 %
câmera traseira	536	99,08 %
memória interna	532	98,34 %
sistema operacional	532	98,34 %
sac	532	98,34 %
tamanho do display	529	97,78 %
referência do modelo	529	97,78 %
processador	521	96,30 %
tipo de chip	521	96,30 %
cor	515	95,19 %
expansivo até	494	91,31 %
câmera frontal	493	91,13 %
modelo	486	89,83 %
resolução	482	89,09 %

Tabela 14 – Propriedades menos frequentes da ficha técnica em toda a base B1.

atributo	#	%
marca	477	88,17 %
memória ram	476	87,99 %
tipo de tela	467	86,32 %
recursos de chamada	457	84,47 %
nfc	443	81,89 %
peso líq aproximado do produto kg	335	61,92 %
fornecedor	335	61,92 %
dimensões aproximadas do produto cm axlxp	335	61,92 %
quantidade de chips	313	57,86 %
conectividade	308	56,93 %
tv	299	55,27 %
versão	268	49,54 %
fabricante	206	38,08 %
versão s o	205	37,89 %
multichip	205	37,89 %
sintonizador de tv	205	37,89 %
conexões	201	37,15 %
dimensões do produto cm axlxp	197	36,41 %
peso liq aproximado do produto kg	195	36,04 %
outros recursos	88	16,27 %

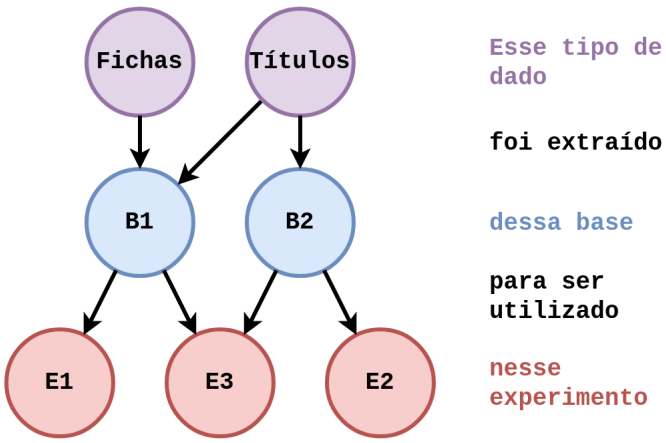
Figura 21 – Exemplo de título anotado da base B2.

{'text': '**smartphone samsung galaxy j7 prime dual chip android 6.0 tela 5.5 32gb 4g câmera 13mp**',
'tokens': [
{'text': 'smartphone','start': 0,'end': 10},
{'text': 'samsung', 'start': 11, 'end': 18},
{'text': 'galaxy', 'start': 19, 'end': 25},
(...)],

'spans': [
{'start': 0,'end': 10, 'label': 'WIT'},
{'start': 11,'end': 18, 'label': 'MARCA'},
(...)]}

A partir dos dados de B1 e B2, três experimentos foram realizados, conforme resumido na Figura 22.

Figura 22 – Diagrama explicativo do uso das bases de dados nos experimentos e quais tipos de dados foram utilizados em cada experimento.



1. O primeiro experimento (**E1**) teve como objetivo verificar a quantidade de informação útil que pode ser extraída de dados estruturados. Para tanto, triplas SPO foram construídas utilizando informações da base B1. A entidade sujeito é o valor do atributo `modelo` correspondente ao item; a relação é o rótulo de outro atributo; e a entidade objeto, o valor desse atributo. As relações foram estabelecidas de acordo com as entidades nomeadas de B2. Essa modelagem está apresentada nas Figuras 23 e 24.

Figura 23 – Construção de relações das fichas técnicas.

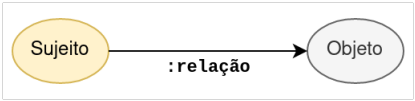
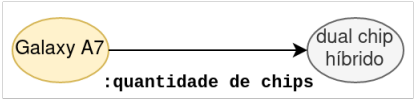


Figura 24 – Exemplo de instância de relação extraída.



Esse experimento resultou em 2.825 triplas modelo-atributo-valor extraídas de B1, divididas entre as respectivas relações conforme mostra a Tabela 15. A Tabela 16 mostra alguns exemplos dessas instâncias.

Tabela 15 – Quantidade de instâncias extraídas de dados estruturados da base B1 via experimento E1. Cada entidade nomeada de B2, diferente de `modelo`, possui uma relação correspondente sendo o objeto do sujeito `modelo`, como explicam as Figuras 23 e 24.

	# instâncias extraídas
`has_color`	476
`has_camera`	348
`has_internal_memory`	348
`has_processor`	344
`has_display_size`	337
`has_os`	335
`has_chip_capacity`	327
`has_brand`	310
TOTAL	2.825

Tabela 16 – Exemplos de triplas extraídas da base B1 via experimento E1.

Relação	**Sujeito**	**Objeto**
`has_internal_memory`	SM-N975F/2DL	256gb
`has_color`	ZC554KL-4A115BR	preto
`has_display_size`	Galaxy S8	5.8”
`has_camera`	Moto G (3ª Geração)	13mp

Fonte: adaptada de (BARBIRATO; REAL; CASELI, 2021)

2. Já o segundo experimento (**E2**), teve como objetivo extrair relações de dados não estruturados utilizando BERT (SOARES et al., 2019). Para isso, utilizou-se os títulos anotados da base B2, cujos conjuntos de treino, validação e teste foram divididos conforme mostra a Tabela 17. Esses títulos foram adaptados para servir de entrada para o método, como ilustra a Tabela 18.

   O método de Soares et al. (2019) foi adaptado[6] para dois modelos BERT capazes de lidar com o português do Brasil: o Multilingual BERT (mBERT) (DEVLIN et al., 2019) e o BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020). Desses, o modelo treinado com o BERTimbau obteve melhor performance (F1), extraindo corretamente 378 instâncias de relações (triplas), exemplificadas na Tabela 19.

[6] <https://github.com/joaobarbirato/BERT-Relation-Extraction>

Tabela 17 – Divisão de treino, validação e teste da base B2 para a realização do experimento E2.

	treino	validação	teste	**total**
`has_brand`	199	103	103	405
`has_camera`	108	70	53	231
`has_chip_capacity`	124	63	66	253
`has_color`	170	89	92	351
`has_display_size`	117	67	68	252
`has_internal_memory`	127	77	73	277
`has_os`	68	39	40	147
`has_processor`	18	9	8	35
Total	931	517	503	1951

Fonte: adaptado de (BARBIRATO; REAL; CASELI, 2021).

Tabela 18 – Exemplo de um título da base B2, processado para servir de entrada para o método de Soares et al. (2019) no experimento E2. No exemplo, as duas entidades são evidenciadas por anotação automática de casamento exato. Além disso, a relação entre elas também é posta logo abaixo.

Título original	**Título processado para o método**
smartphone ms60f tela 5,5 1gb ram android 7 multilaser	smartphone <e1>ms60f</e1> tela 5,5 <e2>1gb</e2> ram android 7 multilaser `has_internal_memory(e1,e2)`

Tabela 19 – Exemplos de relações extraídas no experimento E2 usando o BERTimbau. A segunda coluna representa a instância verdadeira de relação presente na sentença da primeira coluna, enquanto que a terceira mostra a instância predita pelo modelo.

Sentença	**Verdadeira**	**Predita**
smartphone **[E1]akua[/E1]** **[E2]ek4[/E2]** dual sim 3g tela 4.0” 4gb câm 5mp branco.	`has_brand(e2,e1)`	`has_brand(e2,e1)`
smartphone blu **[E1]grand m[/E1]** dual sim 3g 5.0"5mp 3.2mp **[E2]cinza[/E2]**	`has_color(e1,e2)`	`has_color(e1,e2)`
celular smartphone ms5 colors 4,5 **[E1]branco[/E1]** **[E2]p3311[/E2]** multilaser	`has_color(e2,e1)`	`has_brand(e2,e1)`

Fonte: adaptado e estendido de (BARBIRATO; REAL; CASELI, 2021).

Em seguida, utilizou-se o modelo treinado em E2 – partindo de dados não estruturados anotados de B2 – para inferir relações nos títulos de B1, uma base diferente. Assim, foi possível comparar as informações extraídas partindo-se de uma mesma base, porém de naturezas diferentes, em E3.

Para que os títulos de B1 servissem de entrada, utilizou-se um modelo Reconhecedor de Entidade Nomeada (REN) treinado no domínio do e-commerce (SILVA et al., 2021). Sua aplicação resultou em 4.933 títulos marcados para servir de entrada para o modelo de E2. Assim, o modelo BERTimbau inferiu 2.072 triplas nesses títulos. A Tabela 20 resume as medidas de avaliação obtidas em E2.

Tabela 20 – Medidas de avaliação em dados não estruturados da base (a) B2 e (b) B1.

		(a) **B2**				(b) **B1**		
		BERTimbau		**mBERT**			**BERTimbau**	
Relação	**# instâncias**	**Acurácia**	**F1**	**Acurácia**	**F1**	**# instâncias**	**Acurácia**	**F1**
`has_processor`	8	87,50	**93,33**	62,50	66,67	476	50,84	66,30
`has_os`	40	90,00	92,31	90,00	**93,51**	605	77,85	79,63
`has_internal_memory`	73	100,00	97,99	100,00	**99,32**	15	80,00	4,57
`has_display_size`	68	89,71	94,57	92,65	**96,18**	759	74,18	83,90
`has_color`	92	98,91	**94,79**	97,83	91,37	645	94,26	84,04
`has_chip_capacity`	66	89,39	89,39	92,42	**93,85**	589	78,95	84,16
`has_camera`	53	100,00	**96,36**	100,00	95,50	1101	92,28	93,81
`has_brand`	103	90,29	**92,08**	85,44	87,13	743	75,24	81,84
$\texttt{Média}_{micro}$	-	93,23	**93,85**	90,10	90,44	-	77,95	72,28

Fonte: adaptada de (BARBIRATO; REAL; CASELI, 2021)

3. Por fim, o terceiro experimento (**E3**) consistiu em comparar as quantidades de informações extraídas de ambos os tipos de dados. Comparando dados de bases diferentes, é possível entender quão diferentes (e possivelmente complementares) essas informações podem ser. Concluiu-se, com E3, que cerca de 97% das triplas extraídas dos títulos de B2 são novas e diferentes do conjunto de triplas extraídas em E1. Da mesma forma, comparando dados diferentes de uma mesma base, a B1, é possível entender quão complementares essas informações podem ser. Além disso, constatou-se que cerca de 90% das triplas extraídas dos títulos de B1 também são novas e diferentes do conjunto de triplas extraídas em E1.

A partir dos experimentos descritos nesta seção, apresentados em detalhes em (BARBIRATO; REAL; CASELI, 2021), concluiu-se que extrair informações de dados não estruturados é uma tarefa promissora para o domínio do e-commerce. Destaca-se que esses dados são mais abundantes e fáceis de coletar do que os dados estruturados das fichas técnicas de produtos. Essa constatação embasa o restante do trabalho e responde a pergunta colocada no título desta seção.

4.3.1.2 Extração de triplas

Esta seção descreve como as informações nos dados não estruturados foram extraídas e utilizadas para construir uma base de triplas. Essa base tem como objetivo popular os KGs descritos em 4.3.2.

A ideia é construir um KG utilizando produtos e suas informações não-estruturadas que os caracterizam como entidades. Assim, cada sintagma listado na Tabela 12 foi procurado em títulos e descrições de produto. Caso o produto tivesse um deles, uma tripla de relação `has_feature` era criada entre o identificador do produto (entidade sujeito) e o sintagma encontrado em seus dados não estruturados (entidade objeto). A Tabela 21 traz exemplos de triplas (item, relação, característica) extraídos seguindo essa estratégia.

A extração resultou em 1.495 triplas da base de itens do tipo Conexão (BC) e 1.346 do tipo Sortimento (BS) – conjuntos de triplas esses que daqui em diante serão denominados

Tabela 21 – Exemplos de triplas extraídas do Córpus Americanas S.A.

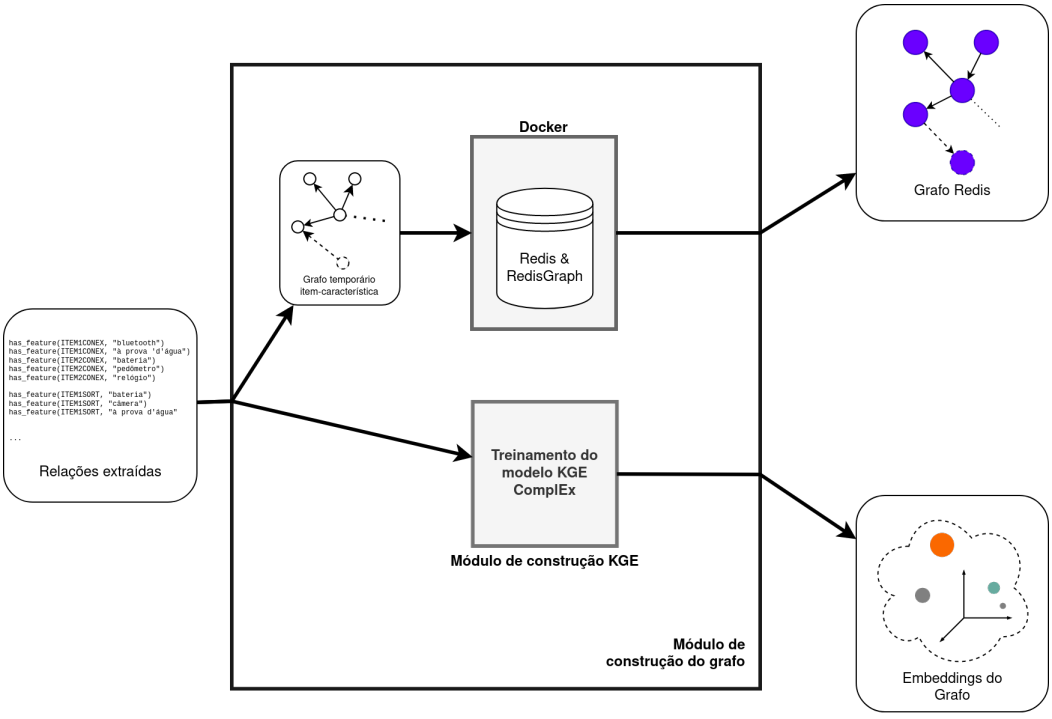
item	relação	característica
ITEM1CONEXAO	has_feature	`bluetooth`
ITEM1CONEXAO	has_feature	`à prova d'água`
ITEM2CONEXAO	has_feature	`pedômetro`
ITEM2SORTIMENTO	has_feature	`cronômetro`

respectivamente como TC e TS.

4.3.2 Módulo de construção do grafo de conhecimento

Nesta seção descreve-se como os grafos de conhecimento no domínio do e-commerce foram construídos utilizando os conjuntos de triplas TC e TS, resultados do módulo descrito na seção 4.3.1.2. A Figura 25 ilustra o funcionamento do módulo de construção de grafo de conhecimento no domínio do e-commerce.

Figura 25 – Diagrama de funcionamento do módulo de construção de KGs. As relações extraídas do módulo em 4.3.1 servem para construir o Grafo Redis interagindo com o banco Redis e sua ferramenta RedisGraph; assim como também servem para treinar o modelo KGE ComplEx (TROUILLON et al., 2016) parar criar representações KGEs.



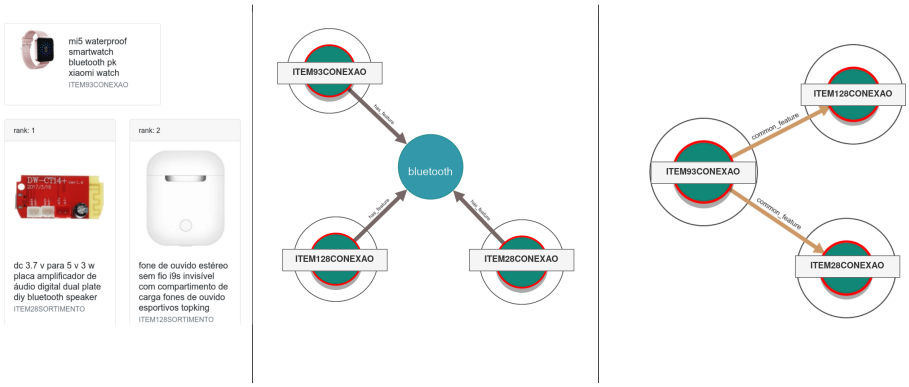
A seção 4.3.2.1 descreve a construção de um KG utilizando a ferramenta RedisGraph; enquanto que a seção 4.3.2.2 explica o método que constrói um grafo a partir de representações no espaço de *embeddings* (KGE).

4.3.2.1 Construção do grafo no RedisGraph

Partindo-se de atributos e valores correspondentes, constrói-se um grafo com triplas (h, r, t), onde h é um item (especificamente diferenciado pelo seu ID), r é a relação `tem_característica` e t é a característica encontrada na descrição, ou no título, do produto. Assim, é possível descobrir quais dois itens possuem uma mesma relação que os ligam à mesma caraterística.

Por exemplo, se um determinado item de BC possui a característica `touchscreen` e um item de BS também possui essa característica, é possível inferir que existe uma relação ligando diretamente ambos por meio da característica em comum entre eles: `touchscreen`. O resultado disso é um grafo populado por triplas (h, r, t), onde h e t são produtos e r é um rótulo de característica em comum entre ambos os produtos. Essa ideia está exemplificada na Figura 26.

Figura 26 – A parte (a) mostra três produtos disponíveis nas plataformas de e-commerce da companhia parceira. (b) os produtos $ITEM93CONEXAO$, $ITEM28SORTIMENTO$ e $ITEM128SORTIMENTO$ possuem, por exemplo, a mesma característica – `bluetooth` – evidenciada no grafo pela relação `has_feature`. Portanto, (c) cria-se um grafo que liga diretamente esses 3 itens por características em comum.



No problema de recomendação de produtos, que é o investigado neste trabalho, enquanto o usuário olha a página de um produto, espera-se que um outro produto seja recomendado a ele. Assim, com o grafo de produtos construído, a estratégia adotada para oferecer recomendação se baseia em dois fatores:

1. **Preço do item** – quanto menor for o preço do produto, maior a chance de ele ser recomendado.

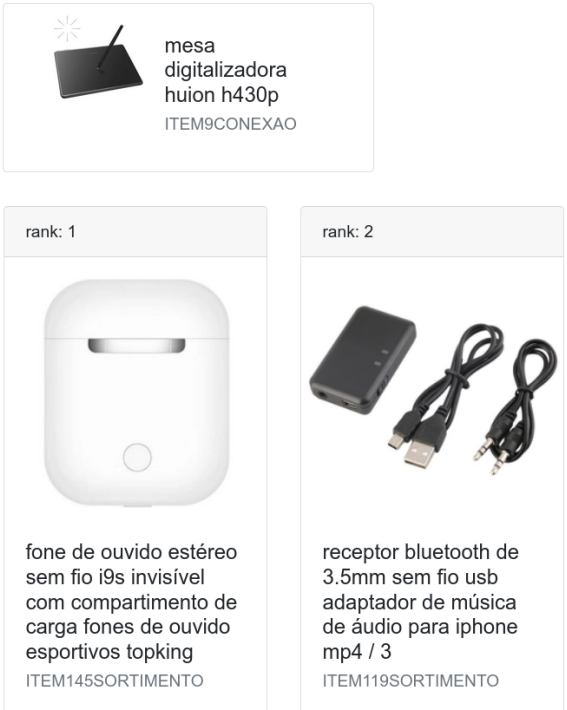
2. **Características em comum entre produtos** – quanto maior for a soma dos pesos das ligações entre o item recomendado e o item observado, maior a chance de ele ser recomendado.

Então, recomenda-se os vizinhos a um salto – caminho por uma aresta que liga duas entidades diretamente – de distância, sendo o ranking de produtos feito a partir do cálculo da pontuação (score) apresentada na Equação 14. Os preços foram normalizados em uma escala de 1 a 2, de acordo com o tipo do produto. Assim, considere que um usuário visualiza um produto p_h e lhe é recomendado um outro produto p_t para o qual existem arestas $e_{h,t} \in E_{h,t}$ que ligam ambos. Então, o score é calculado segundo a Equação 14.

$$score(p_h, p_t) = \sum_{e_{h,t} \in E_{h,t}} \frac{\text{peso\_características\_em\_comum}(e_{h,t})}{\text{preço}(p_t)} \tag{14}$$

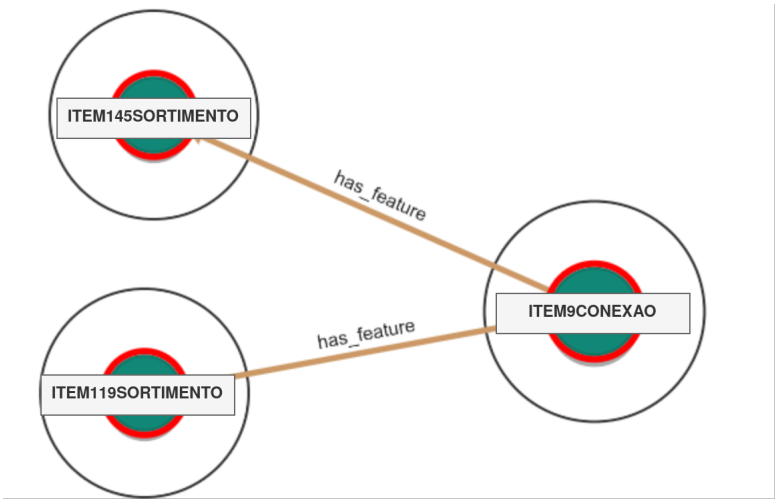
Utilizando itens de TC e TS como observados e recomendados, respectivamente, a aplicação da estratégia de pontuação resultou em 883 recomendações. Dessas, são 15 itens observados diferentes que levam a outros 107 itens recomendados diferentes. Em média, cada item observado recomenda 58, 867 outros produtos. As Figuras 27 e 28 exemplificam, com os mesmos três produtos (`ITEM9CONEXAO`, `ITEM145SORTIMENTO` e `ITEM119SORTIMENTO`, a recomendação realizada por esse método.

Figura 27 – Exemplo de recomendação realizada pelo KG com RedisGraph. O item observado nesta figura é o `ITEM9CONEXAO`, enquanto que `ITEM145SORTIMENTO` e `ITEM119SORTIMENTO` são itens recomendados.



Esses mesmos 15 itens observados serão utilizados para comparar com o método em 4.3.2.2.

Figura 28 – Exemplo de recomendação realizada pelo KG com RedisGraph. Tratam-se dos mesmos itens presentes na Figura 27 visualizados na plataforma RedisInsight.



4.3.2.2 Construção de KGE

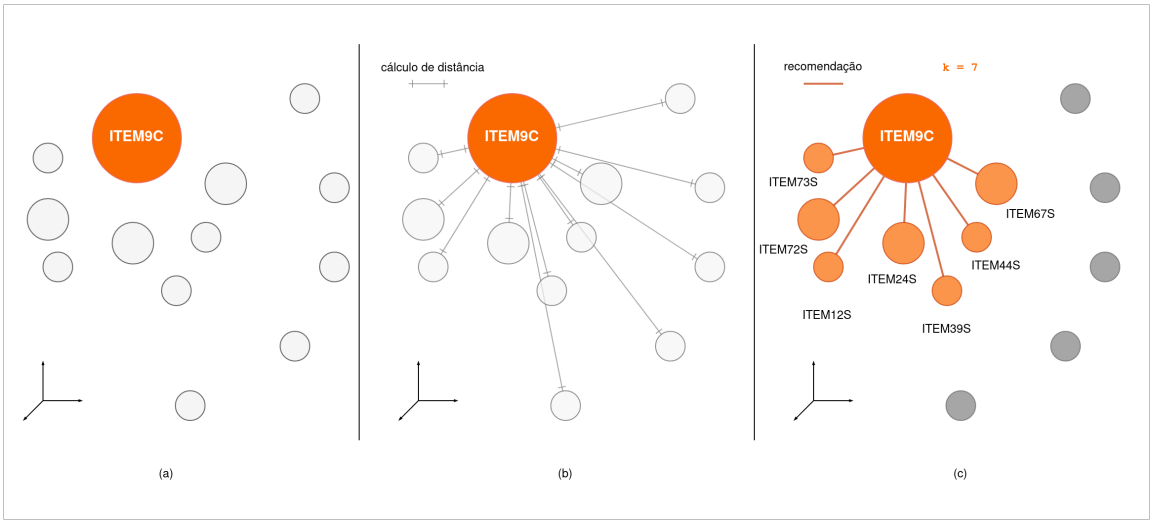
Métodos KGE representam elementos de grafos em um espaço denso de *embeddings* (veja seção 2.1.1). Assim, essa seção visa detalhar como as representações de TC e TS foram construídas neste espaço.

As *embeddings* foram treinadas pelos métodos descritos na seção 3.1: TransE (BORDES et al., 2013b) e ComplEx (TROUILLON et al., 2016). O uso do primeiro justifica-se pelo fato das relações não serem simétricas; enquanto que o segundo foi utilizado por ser um método sofisticado de *matching* semântico e também por já ter sido usado no domínio do e-commerce (XU et al., 2020). Suas implementações foram utilizadas da biblioteca Ampligraph (COSTABELLO et al., 2019). A seguir são apresentadas algumas informações sobre o treinamento desses modelos:

- Os modelos foram treinados com *holdout* de 70% das triplas para treino, 20% para teste e 10% para validação.
- O treinamento foi feito em um processador Intel Core i71185G7, 3GHz.
- Ambos os modelos foram treinados com 20 épocas, 150 dimensões de *embeddings* e otimizador Adam com taxa de aprendizado de 10^3, como sugerido na documentação da biblioteca Ampligraph.
- O modelo ComplEx atingiu, no conjunto de teste, MMR e Hits@10 de $0,840$ e $0,886$ respectivamente, enquanto que o modelo TransE, $0,266$ e $0,623$.
- Pela diferença de métricas, escolheu-se o modelo ComplEx para prosseguir com os experimentos.

Com o modelo de *embeddings* treinado, utilizou-se a estratégia de k vizinhos mais próximos para encontrar entidades vizinhas. No contexto da recomendação de produtos, o intuito é encontrar produtos com características latentes parecidas, ou seja, características que não são triviais para um humano. Portanto, a recomendação é baseada na distância entre duas entidades no espaço de *embeddings*. A Figura 29 ilustra essa estratégia. A parte (a) ilustra entidades no espaço de *embeddings*. Em destaque, laranja, está a entidade `ITEM9C`, representando um item observado. Assim, a estratégia de vizinhos mais próximos (b) visa calcular a distância entre os exemplos (entidades, nós de um grafo, itens) e selecionar os mais próximos ao item observado. Por fim, selecionam-se, nesse caso, os 7 vizinhos mais próximos (c) como itens recomendados, os quais no exemplo são `ITEM73S`, `ITEM72S`, `ITEM12S`, `ITEM24S`, `ITEM39S`, `ITEM44S` e `ITEM67S`.

Figura 29 – Ilustração de como a estratégia de vizinhos mais próximos funciona para KGEs.



Capítulo 5

Resultados

Neste capítulo serão descritos os resultados dos métodos detalhados no capítulo 4. A seção 5.1 traz os resultados quantitativos, ou seja, compara a quantidade de recomendações geradas pelo grafo no RedisGraph (RR) e de recomendações geradas pelas representações KGE (RE); enquanto que a seção 5.2 mostra resultados qualitativos, ou seja, busca entender algumas recomendações de RE.

5.1 Análise quantitativa

No intuito de comparar RR e RE, foi necessário estabelecer o parâmetro que indica quantos vizinhos mais próximos as *embeddings* seriam recuperados (k). Para isso, estudou-se qual valor de k seria mais adequado para comparar os dois conjuntos.

Duas quantidades interessantes de se estudar também são as diferenças de conjunto entre RR e RE, ou seja, a quantidade de recomendações exclusivamente em RR (XR) e em RE (XE). As Equações 15 e 16 ilustram a ideia dessas medidas.

$$XR = RR - RE \tag{15}$$

$$XE = RE - RR \tag{16}$$

A Figura 30 mostra um gráfico das porcentagens de pares de entidades em XR e em XE (eixo vertical) em função da quantidade k de vizinhos recomendados por item observado (eixo vertical). O intuito desse gráfico é representar a cobertura de itens recomendados. Ambas as linhas se cruzam entre 30 e 40% (ou seja, quando k está perto de 60 vizinhos),

situação onde ambas possuem 60 a 30% de seus pares observado-recomendado na intersecção entre XR e XE. É possível ver que o gráfico correspondente a XR é descendente, o que indica que quanto, maior o k, mais semelhantes são os elementos de RE em relação a RR.

Entretanto, por se tratar de uma abordagem que depende da distância no espaço, maiores valores de k resultam em recomendações de itens com características latentes mais distintas. Por isso, a Figura 31 mostra o gráfico da distância euclidiana média entre o item observado e todos seus k vizinhos (eixo vertical) em função do valor de k (eixo horizontal).

Figura 30 – Cobertura dos itens recomendados em função do valor de k (horizontal).

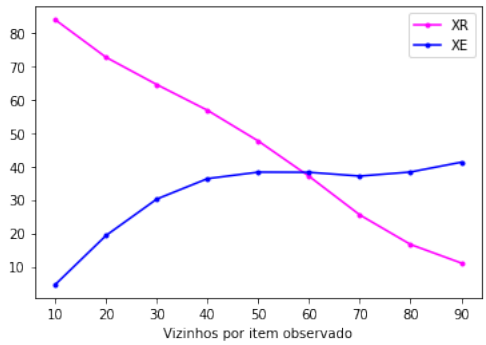
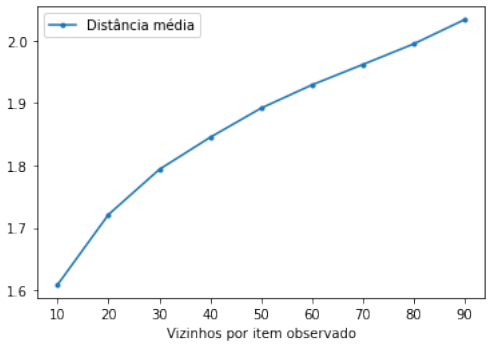


Figura 31 – Distância média das triplas cujas entidades foram recomendadas em função do valor de k.



Com base no gráfico da Figura 30, sabe-se que a média de recomendações por item distinto no RedisGraph foi de aproximadamente 60 (58, 867). Logo, definiu-se como quantitativamente razoável fixar k como 60. Além disso, a partir do gráfico da Figura 31, entende-se que entre $k = 10$ e $k = 60$, a distância média aumentou 16, 648% e, portanto, o algoritmo de vizinhos mais próximos recuperou entidades cerca de 17% mais distantes ao fixar $k = 60$.

Nessa configuração que considera 60 vizinhos, RE contém 900 recomendações. Delas, XE representa 344, ou seja 38,22% delas são exclusivas do KGE e consequentemente, 556 aparecem tanto em RE quanto em RR. Dos 15 itens observados, 11 deles recomendaram mais itens diferentes que a recomendação RR. Dois desses 11 observados, e alguns respectivos itens recomendados, estão listados na Tabela 22 e exemplificados nas Figuras 32 e 33.

Os 344 itens recomendados de XE serão analisados na seção 5.2.

5.2 Análise qualitativa

Esta seção busca entender as recomendações feitas pelo método KGE descrito em 4.3.2.2. Um dos problemas das propriedades latentes é sua interpretabilidade pelos humanos.

Tabela 22 – Recomendações a partir dos itens observados `ITEM93CONEXAO` e `ITEM9CONEXAO`. A segunda linha mostra recomendações exclusivas do método utilizando Redis descrito em 4.3.2.1, enquanto que a terceira mostra exclusivas do método utilizando KGE (4.3.2.2).

item observado	ITEM93CONEXAO	ITEM9CONEXAO
2 itens recomendados XR	ITEM40SORTIMENTO, ITEM113SORTIMENTO	ITEM64SORTIMENTO, ITEM5SORTIMENTO
2 itens recomendados XE	ITEM61SORTIMENTO, ITEM100SORTIMENTO	ITEM19SORTIMENTO, ITEM23SORTIMENTO

Figura 32 – Recomendações geradas a partir do item `ITEM93CONEXAO`. À esquerda, em roxo, estão itens recomendados que pertencem à XR, enquanto que à direita, em laranja, estão os que pertencem à XE.

Fontes: próprio autor e <https://www.americanas.com.br/>. Último acesso: 8 de abril de 2022.

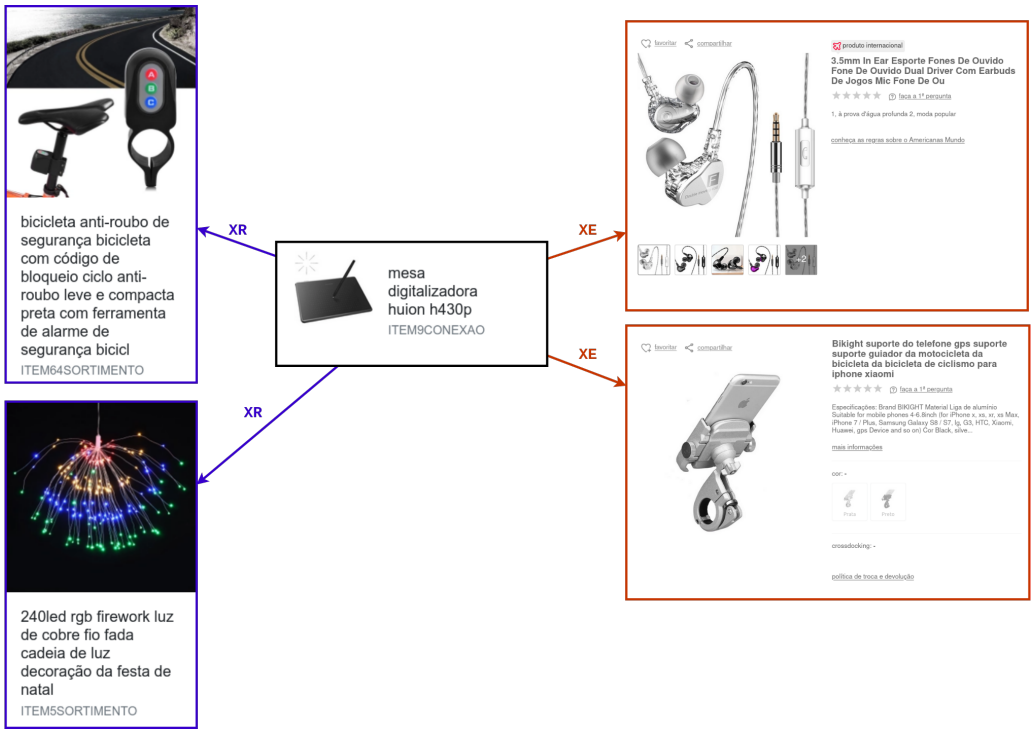
Nesse contexto, esta seção analisa os 344 itens recomendados do conjunto XE, evidenciados na seção anterior.

Dos 15 itens observados, 1 deles é um `tablet`, 10 deles são `mesas digitalizadoras` e 4 deles são `smartwatches`. A partir desses 15 produtos 67 itens diferentes foram recomendados. Cada um deles foi avaliado considerando-se o seguinte:

"`Partindo do item observado, a recomendação do item recomendado é ou não esperada?`"

Ou seja, quais recomendações os clientes da plataforma de e-commerce esperariam que ocorressem ao observar o item. Por exemplo, ao observar um *smartwatch*, é esperado que

Figura 33 – Recomendações geradas a partir do item `ITEM9CONEXAO`. À esquerda, em roxo, estão itens recomendados que pertencem à XR, enquanto que à direita, em laranja, estão os que pertencem à XE.



Fontes: próprio autor e <https://www.americanas.com.br/>. Último acesso: 8 de abril de 2022.

sejam recomendados itens do mesmo departamento, ou ainda, do mesmo nicho de produtos tecnológicos, como fones de ouvido e *smartphones*; mas não é esperado que um brinquedo ou um utensílio doméstico seja recomendado. Portanto, caso o item recomendado fosse, de fato, recomendado, seria avaliado como correto. Caso contrário, incorreto.

Para tanto, foram identificadas as WITs dos itens recomendados e a avaliação foi feita considerandos-se a pergunta anterior e as WITs do item observado e do item recomendado. A Tabela 23 mostra quais padrões WIT de item observado - WIT de item recomendado foram encontrados nos itens avaliados e justificativas.

Das 344 recomendações, 315 delas $(91,57\%)$ não foram esperadas e, consequentemente, 29 $(8,43\%)$ foram. As 29 recomendações esperadas – além das 400 presentes na intersecção entre RR e RE – mostram que o uso de métodos KGE ainda é capaz de trazer recomendações explicáveis como o método Redis. Além disso, as outras 315 mostram que a exploração dos atributos latentes das *embeddings* de grafo podem contribuir para problemas de negócio que visam certa diversidade de produtos, uma vez que os itens recomendados são, neste contexto, semanticamente similares porém não são os mais óbvios para a tarefa de recomendação.

Tabela 23 – À esquerda estão a avaliação e sua justificativa. À direita, estão os respectivos padrões de WITs na recomendação .

avaliação e justificativa	**WITs recomendadas se a WIT observada for**		
	mesa digitalizadora	**tablet**	**smartwatch**
incorreto, muito diferente	placa amplificadora de audio led livro relógio estátua jardim acessório relógio adesivo capa smartphone babador controle remoto acessório gps brinquedo acessório carro mesa cabeceira acessório smartphone acessório camera suporte joia cortina de chuveiro tenis quadro tatuagem pingente	 led adesivo cortina de chuveiro tenis tatuagem	placa amplificadora de audio led estátua jardim adesivo mouse gamer auto falante brinquedo mesa cabeceira medidor gordura acessório tatuagem purificador acessório pets cofre acessório camera lanterna suporte joia repelente elétrico cortina de chuveiro tenis tatuagem pingente natal
incorreto, diferente	relógio pulso fone de ouvido	-	relógio pulso
correto, finalidade parecida	mouse gamer	-	capa smartphone
correto, parecido	adaptador cabo webcam	-	acessório relógio
correto, Lifestyle	-	-	fone de ouvido
correto, caracteristicas em comum	led fone de ouvido ventilador	-	-

Capítulo 6

Conclusões

Este trabalho abordou a construção de grafos de conhecimento para o domínio do e-commerce, onde os vértices representam produtos e suas características, enquanto que as relações que os ligam descrevem se o produto possui tal característica ou não. Para tanto, estudou-se a viabilidade da extração de informação em dados não estruturados nesse domínio.

Além disso, a principal contribuição deste trabalho foi a implementação de dois métodos: um distributivo, baseado nos recursos do RedisGraph; e outro distribuído, fundamentado em *embeddings* de grafos de conhecimento. Ambos foram avaliados em uma base de dados não estruturados no domínio estudado, em português do Brasil, com o intuito de estudar a efetividade de métodos KGE na tarefa de recomendação de produtos.

Os resultados evidenciam que métodos KGE trazem não apenas recomendações parecidas com as do método distributivo, mas também outras não esperadas porém baseadas na similaridade semântica dos elementos do grafo. Esse fato faz-se coerente com a resolução de problemas no domínio do e-commerce, como a cauda longa.

Como principais desdobramentos deste trabalho, destacam-se: (i) o estudo de outros nichos de produtos; (ii) a investigação de diferentes usos para a função *score*, no contexto de KGE e do e-commerce, como medida de similaridade; e (iii) a geração de regras que expliquem as inferências produzidas pelos métodos KGE.

Em relação ao primeiro, este trabalho mostrou experimentos referentes aos itens que compartilham do mesmo "Mundo Conexão". Entretanto, a empresa parceira possui diversos outros mundos, como "Mundo Casa", "Mundo Automotivo" e "Mundo Reforma"[1]. Sabe-se que a generalização de modelos no domínio do e-commerce é um desafio (XU et

[1] <https://www.americanas.com.br/hotsite/americanas-mundo>. Último acesso: 13 de abril de 2022.

al., 2020) e, portanto, tal investigação seria interessante para verificar se categorias de produtos diferentes podem produzir comportamentos diferentes.

Entretanto, embora este trabalho utilize dados do domínio do e-commerce, os métodos propostos não são dependentes do domínio. Bastam métodos para obter relações entre características de entidades para que a ferramenta Redisgraph seja utilizada e o modelo seja treinado. Por exemplo, é possível atuar no domínio de culinária, ligando receitas e ingredientes; no contexto de trabalhos acadêmicos, autores e tópicos; em filmes, cineastas e gêneros. Em todos esses domínios, é possível explorar recomendações baseadas somente nas características das entidades em questão.

No contexto de KGE, a similaridade entre duas entidades pode consistir no valor da função *score*. Abordagens de *Matching* Semântico (descritas em 2.2.2), como o HolE (NICKEL et al., 2016) e o ComplEx (TROUILLON et al., 2016) baseiam-se na similaridade para verificar a verossimilhança de um fato. Por exemplo, o método HolE baseia-se na função *score* $f_r(h,t) = r^\top(h \star t)$, onde $\star$ é a correlação circular[2]. Nesse contexto, é possível mensurar a similaridade entre dois itens comparando valores de f_r para um conjunto de relações ou características de interesse R. Por exemplo, é possível determinar se dois itens são parecidos comparando combinação de f_{cor}, f_{marca} e f_{SO}, para cada item. Assim, investigar usos para a função *score* no domínio do e-commerce é outra possível direção para trabalhos futuros.

Por fim, o uso excessivo de métodos contextuais, como o BERT (DEVLIN et al., 2019), conforme apontado por Bommasani et al. (2021),ntraz problemas de generalização. Métodos distribuídos não estão distantes dessa problemática, uma vez que também estão ancorados na mesma fundamentação de abstrair características de instâncias para um espaço vetorial. Assim, uma frente de trabalho decorrente dessa problemática é a de uso de métodos de geração de regras em KGs (ZHANG et al., 2019; SADEGHIAN et al., 2019; LAJUS; GALáRRAGA; SUCHANEK, 2020; ZHANG et al., 2020; ZHANG et al., 2021) para produzir explicações para as inferências feitas a partir de métodos KGE.

[2] Nickel et al. (2016) definem correlação circular como como $[h \star t]_k = \sum_{i=1}^{d} h_i t_{(k+i) mod(d)}$, onde d é a dimensão das *embeddings*.

Referências

ANDRZEJEWSKI, D.; ZHU, X.; CRAVEN, M. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In: **Proceedings of the 26th Annual International Conference on Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2009. (ICML '09), p. 25–32. ISBN 9781605585161. Disponível em: <https://doi.org/10.1145/1553374.1553378>.

BARBIRATO, J.; REAL, L.; CASELI, H. Relation extraction in structured and unstructured data: a comparative investigation on smartphone titles in the e-commerce domain. In: **Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana**. Porto Alegre, RS, Brasil: SBC, 2021. p. 101–110. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/stil/article/view/17789>.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **J. Mach. Learn. Res.**, JMLR.org, v. 3, n. null, p. 993–1022, mar. 2003. ISSN 1532-4435.

BOLLACKER, K. et al. Freebase: A collaboratively created graph database for structuring human knowledge. In: **Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2008. (SIGMOD '08), p. 1247–1250. ISBN 9781605581026. Disponível em: <https://doi.org/10.1145/1376616.1376746>.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, 2021.

BORDES, A.; CHOPRA, S.; WESTON, J. Question answering with subgraph embeddings. **arXiv preprint arXiv:1406.3676**, 2014.

BORDES, A. et al. Irreflexive and hierarchical relations as translations. **arXiv preprint arXiv:1304.7158**, 2013.

_____. Translating embeddings for modeling multi-relational data. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems 26**. Curran Associates, Inc., 2013. p. 2787–2795. Disponível em: <http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf>.

BORDES, A.; WESTON, J.; USUNIER, N. Open question answering with weakly supervised embedding models. In: CALDERS, T. et al. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 165–180. ISBN 978-3-662-44848-9.

CAILLIAU, P. et al. Redisgraph graphblas enabled graph database. In: IEEE. **2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)**. [S.l.], 2019. p. 285–286.

COSTABELLO, L. et al. **AmpliGraph: a Library for Representation Learning on Knowledge Graphs**. 2019. Disponível em: <https://doi.org/10.5281/zenodo.2595043>.

DETTMERS, T. et al. Convolutional 2d knowledge graph embeddings. In: **Proceedings of the 32th AAAI Conference on Artificial Intelligence**. [s.n.], 2018. p. 1811–1818. Disponível em: <https://arxiv.org/abs/1707.01476>.

DEVLIN, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://www.aclweb.org/anthology/N19-1423>.

DIGITAIS, R. **Dados de ecommerce no Brasil: confira os principais números do comércio eletrônico**. 2021. <https://resultadosdigitais.com.br/marketing/dados-de-ecommerce-no-brasil/>.

DOMO. **Data Never Sleeps 6.0**. 2018. <https://www.domo.com/solution/data-never-sleeps-6>.

DUVENAUD, D. et al. Convolutional networks on graphs for learning molecular fingerprints. In: . [s.n.], 2015. abs/1509.09292. Disponível em: <http://arxiv.org/abs/1509.09292>.

HAN, X. et al. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. **arXiv preprint arXiv:1810.10147**, 2018.

HENDRICKX, I. et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. **arXiv preprint arXiv:1911.10422**, 2019.

JI, G. et al. Knowledge graph embedding via dynamic mapping matrix. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. Beijing, China: Association for Computational Linguistics, 2015. p. 687–696. Disponível em: <https://www.aclweb.org/anthology/P15-1067>.

_____. Knowledge graph embedding via dynamic mapping matrix. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. Beijing, China: Association for Computational Linguistics, 2015. p. 687–696. Disponível em: <https://www.aclweb.org/anthology/P15-1067>.

_____. Knowledge graph completion with adaptive sparse transfer matrix. In: **Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2016. (AAAI'16), p. 985–991.

JIANG, S. et al. Towards the completion of a domain-specific knowledge base with emerging query terms. In: **2019 IEEE 35th International Conference on Data Engineering (ICDE)**. [S.l.: s.n.], 2019. p. 1430–1441.

JIANG, X. et al. Relation extraction with multi-instance multi-label convolutional neural networks. In: **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers**. Osaka, Japan: The COLING 2016 Organizing Committee, 2016. p. 1471–1480. Disponível em: <https://www.aclweb.org/anthology/C16-1139>.

KUTIYANAWALA, A.; VERMA, P. et al. Towards a simplified ontology for better e-commerce search. **arXiv preprint arXiv:1807.02039**, 2018.

LAJUS, J.; GALáRRAGA, L.; SUCHANEK, F. Fast and exact rule mining with AMIE 3. In: HARTH, A. et al. (Ed.). **The Semantic Web**. Springer International Publishing, 2020. v. 12123, p. 36–52. ISBN 978-3-030-49460-5 978-3-030-49461-2. Disponível em: <http://link.springer.com/10.1007/978-3-030-49461-2_3>.

MEDEIROS, M. A. **Webshoppers 42: Ecommerce tem a maior alta em 20 anos**. 2020. <https://ecommercedesucesso.com.br/ecommerce-bate-recorde>.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119.

MILLER, G. A. Wordnet: A lexical database for english. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 38, n. 11, p. 39–41, nov. 1995. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/219717.219748>.

MINTZ, M. et al. Distant supervision for relation extraction without labeled data. In: **Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP**. Suntec, Singapore: Association for Computational Linguistics, 2009. p. 1003–1011. Disponível em: <https://www.aclweb.org/anthology/P09-1113>.

NGUYEN, D. Q. et al. A novel embedding model for knowledge base completion based on convolutional neural network. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 327–333. Disponível em: <https://www.aclweb.org/anthology/N18-2053>.

NICKEL, M.; KIELA, D. Poincaré embeddings for learning hierarchical representations. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 6338–6347.

NICKEL, M. et al. Holographic embeddings of knowledge graphs. In: **AAAI**. [S.l.: s.n.], 2016. v. 2, n. 1, p. 3–2.

NICKEL, M.; TRESP, V.; KRIEGEL, H.-P. A three-way model for collective learning on multi-relational data. In: **Proceedings of the 28th International Conference on International Conference on Machine Learning**. Madison, WI, USA: Omnipress, 2011. (ICML'11), p. 809–816. ISBN 9781450306195.

OLIVEIRA, H. G. et al. As wordnets do português. **Oslo Studies in Language**, v. 7, n. 1, p. 397–424, 2015.

PETERS, M. E. et al. Deep contextualized word representations. **arXiv preprint arXiv:1802.05365**, 2018.

RADFORD, A. et al. Improving language understanding with unsupervised learning. Technical report, OpenAI, 2018.

RIEDEL, S.; YAO, L.; MCCALLUM, A. Modeling relations and their mentions without labeled text. In: BALCÁZAR, J. L. et al. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 148–163. ISBN 978-3-642-15939-8.

RODRIGUES, B. **E-commerce no Brasil: conheça os principais dados, o market share, o crescimento e as principais estatísticas, com atualização mensal!** 2022. <https://www.conversion.com.br/blog/relatorio-ecommerce-mensal/>.

RUFFINELLI, D.; BROSCHEIT, S.; GEMULLA, R. You {can} teach an old dog new tricks! on training knowledge graph embeddings. In: **International Conference on Learning Representations**. [s.n.], 2020. Disponível em: <https://openreview.net/forum?id=BkxSmlBFvr>.

SABOU, M. et al. Learning domain ontologies for web service descriptions: An experiment in bioinformatics. In: **Proceedings of the 14th International Conference on World Wide Web**. New York, NY, USA: Association for Computing Machinery, 2005. (WWW '05), p. 190–198. ISBN 1595930469. Disponível em: <https://doi.org/10.1145/1060745.1060776>.

SADEGHIAN, A. et al. DRUM: End-to-end differentiable rule mining on knowledge graphs. p. 11, 2019.

SANH, V.; WOLF, T.; RUDER, S. A hierarchical multi-task approach for learning embeddings from semantic tasks. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, p. 6949–6956.

SILVA, D. F. et al. Named entity recognition for brazilian portuguese product titles. In: BRITTO, A.; DELGADO, K. V. (Ed.). **Intelligent Systems**. Cham: Springer International Publishing, 2021. p. 526–541. ISBN 978-3-030-91699-2.

SOARES, L. B. et al. Matching the blanks: Distributional similarity for relation learning. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. Florence, Italy: Association for Computational Linguistics, 2019. p. 2895–2905. Disponível em: <https://www.aclweb.org/anthology/P19-1279>.

SOCHER, R. et al. Semantic compositionality through recursive matrix-vector spaces. In: **Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning**. Jeju Island, Korea: Association for Computational Linguistics, 2012. p. 1201–1211. Disponível em: <https://www.aclweb.org/anthology/D12-1110>.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: **9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)**. [S.l.: s.n.], 2020.

SUN, Z. et al. Rotate: Knowledge graph embedding by relational rotation in complex space. **arXiv preprint arXiv:1902.10197**, 2019.

TROUILLON, T. et al. Complex embeddings for simple link prediction. In: **International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2016. v. 48, p. 2071–2080.

VASWANI, A. et al. Attention is all you need. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 5998–6008.

WAN, M. et al. Representing and recommending shopping baskets with complementarity, compatibility, and loyalty. In: ACM. **27th ACM International Conference on Information and Knoweldge Management**. 2018. Disponível em: <https://www.microsoft.com/en-us/research/publication/representing-and-recommending-shopping-baskets-with-complementarity-compatibility-and-loyalty/>.

WANG, Q. et al. Knowledge graph embedding: A survey of approaches and applications. **IEEE Transactions on Knowledge and Data Engineering**, v. 29, n. 12, p. 2724–2743, 2017.

WANG, Z. et al. Knowledge graph embedding by translating on hyperplanes. In: . [s.n.], 2014. Disponível em: <https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531/8546>.

WESTON, J. et al. Connecting language and knowledge bases with embedding models for relation extraction. In: **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**. Seattle, Washington, USA: Association for Computational Linguistics, 2013. p. 1366–1371. Disponível em: <https://www.aclweb.org/anthology/D13-1136>.

XU, B. et al. Metic: Multi-instance entity typing from corpus. In: **Proceedings of the 27th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2018. (CIKM '18), p. 903–912. ISBN 9781450360142. Disponível em: <https://doi.org/10.1145/3269206.3271804>.

_____. Cn-dbpedia: A never-ending chinese knowledge extraction system. In: BENFERHAT, S.; TABIA, K.; ALI, M. (Ed.). **Advances in Artificial Intelligence: From Theory to Practice**. Cham: Springer International Publishing, 2017. p. 428–438. ISBN 978-3-319-60045-1.

XU, D. et al. Product knowledge graph embedding for e-commerce. In: **Proceedings of the 13th International Conference on Web Search and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2020. (WSDM '20), p. 672–680. ISBN 9781450368223. Disponível em: <https://doi.org/10.1145/3336191.3371778>.

YAN, J. et al. A retrospective of knowledge graphs. **Front. Comput. Sci.**, Springer-Verlag, Berlin, Heidelberg, v. 12, n. 1, p. 55–74, fev. 2018. ISSN 2095-2228. Disponível em: <https://doi.org/10.1007/s11704-016-5228-9>.

YANG, B. et al. Embedding entities and relations for learning and inference in knowledge bases. In: **Proceedings of the International Conference on Learning Representations (ICLR) 2015**. [s.n.], 2015. Disponível em: <https://www.microsoft.com/en-us/research/publication/embedding-entities-and-relations-for-learning-and-inference-in-knowledge-bases/>.

YE, Y. et al. Bayes embedding (bem): Refining representation by integrating knowledge graphs and behavior-specific networks. In: **Proceedings of the 28th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2019. (CIKM '19), p. 679–688. ISBN 9781450369763. Disponível em: <https://doi.org/10.1145/3357384.3358014>.

YU, X. et al. Personalized entity recommendation: A heterogeneous information network approach. In: **Proceedings of the 7th ACM International Conference on Web Search and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2014. (WSDM '14), p. 283–292. ISBN 9781450323512. Disponível em: <https://doi.org/10.1145/2556195.2556259>.

ZHANG, F. et al. Collaborative knowledge base embedding for recommender systems. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 353–362. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939673>.

ZHANG, W. et al. Knowledge graph embedding in e-commerce applications: Attentive reasoning, explanations, and transferable rules. In: **The 10th International Joint Conference on Knowledge Graphs**. ACM, 2021. p. 71–79. ISBN 978-1-4503-9565-6. Disponível em: <https://dl.acm.org/doi/10.1145/3502223.3502232>.

_____. Xtranse: Explainable knowledge graph embedding for link prediction with lifestyles in e-commerce. In: WANG, X. et al. (Ed.). **Semantic Technology**. Singapore: Springer Singapore, 2020. p. 78–87. ISBN 978-981-15-3412-6.

_____. Iteratively learning embeddings and rules for knowledge graph reasoning. In: **The World Wide Web Conference on - WWW '19**. ACM Press, 2019. p. 2366–2377. ISBN 978-1-4503-6674-8. Disponível em: <http://dl.acm.org/citation.cfm?doid=3308558.3313612>.

ZHANG, Y. et al. Position-aware attention and supervised data improve slot filling. In: **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2017. p. 35–45.