

Sistemas de Recomendação: abordagens utilizando dados implícitos

Rebeca Andrade Bivar



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2021

Rebeca Andrade Bivar

Sistemas de Recomendação
abordagens utilizando dados implícitos

Monografia apresentada ao curso Ciência da Computação
do Centro de Informática, da Universidade Federal da
Paraíba, como requisito para a obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Prof. Dra. Danielle Rousy Dias Ricarte

Dezembro de 2021

Catálogo na publicação
Seção de Catalogação e Classificação

B624s Bivar, Rebeca Andrade.

Sistemas de recomendação: abordagens utilizando dados implícitos / Rebeca Andrade Bivar. - João Pessoa, 2021.

47 f. : il.

Orientação: Danielle Rousy Dias Ricarte.

TCC (Graduação) - UFPB/CI.

1. Algoritmos. 2. Sistemas de recomendação. 3. Dados implícitos. 4. Preferências de usuários. I. Ricarte, Danielle Rousy Dias. II. Título.

UFPB/CI

CDU 004.021



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado ***Sistemas de Recomendação*** de autoria de Rebeca Andrade Bivar, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Danielle Rousy Dias Ricarte
Universidade Federal da Paraíba

Prof. Dr. Leandro Carlos de Souza
Universidade Federal da Paraíba

Prof. Dr. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Coordenador(a) do Departamento de Informática
Leandro Carlos De Souza
CI/UFPB

João Pessoa, 27 de dezembro de 2021

*"To see a world in a grain of sand
And a heaven in a wild flower
hold infinity in the palm of your hand
And eternity in an hour"*

William Blake

DEDICATÓRIA

Dedico esta monografia a minha queridíssima tia, Marilene, que concluiu sua missão por aqui mais cedo do que esperávamos, deixando muita luz em minha vida e de todos que a conheceram. Tia simboliza mais de 600 mil pessoas que perderam a batalha para o COVID-19. Ela só precisava de uma vacina.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, Maristela Andrade, por não ter medido esforços para me dar uma educação maravilhosa e por ter sido minha maior inspiração. Agradeço também a minha família por todo o suporte, especialmente minhas tias, Amara Andrade, Cristina Andrade, Magna Andrade e Marilene Andrade; meus primos Camila Andrade, por ter me dado a possibilidade de ter realizado essa graduação e por ter me encorajado em diversos momentos, Kleber Andrade, por toda a paciência e disposição em me ouvir, e Victor Andrade, por ter sido meu pilar e minha fonte de energia durante os últimos anos. Agradeço também (e muito) a minha queridíssima avó, Geovanete Andrade, por ser o ser humano mais forte que já conheci e por ter me criado, junto a minha mãe, com tanto amor.

Agradeço também a Amanda Fonseca, Bruna Viana, Cleidson Canel, David Barros, Eulália Aires, Gabriel Belo, Inês Arruda, Igor Correia, Joyce Santos, Mariana Schroeder, Rodrigo Santos, Rodrigo Sarmento, Thaís Castelo Branco e Vitória Maia por todo o apoio que me deram nos últimos anos, os amo de coração.

Não posso esquecer de agradecer aos meus mentores dessa jornada que foi me tornar cientista da computação: Danielle Rousy, minha orientadora, inspiração e amiga, por ter me guiado e valorizado como ninguém; Thaís Gaudêncio, por ter me introduzido a um mundo de oportunidades, e por ter me enxergado nos momentos bons e ruins; e Gilberto Farias, pela mentoria no meu primeiro estágio e pela ajuda nessa jornada.

Agradeço, por último, mas não menos importante, meus não só amigos, como também "parceiros de guerra": Dárcio Basílio, Davi Graça, Douglas Felizardo, Julianny Meireles, Mathews Alves, Mikaelly Felicio, Nadja Araújo e Pedro Abrantes por terem passado manhãs, tardes, noites, e até madrugadas comigo lidando com problemas diversos, por terem feito a graduação ter sido uma das melhores experiências da minha vida e por terem me acolhido tanto quando precisei, serei eternamente grata pela existência de vocês em minha vida. Amo vocês.

RESUMO

Com o crescimento do comércio virtual, a necessidade de mecanismos para personalização da experiência de clientes tem aumentado cada vez mais. Nesse aspecto, o uso de Sistemas de Recomendação tem sido uma das maiores apostas para construir relacionamentos mais fidelizados entre clientes e o comércio. Entretanto, obter recomendações relevantes é uma tarefa pouco trivial, especialmente quando não se é possível obter explicitamente quais as preferências dos clientes. Considerando essa necessidade, essa pesquisa teve como objetivo explorar distintos modelos de recomendação e gerar recomendações com taxas de acerto iguais ou superiores ao estado da arte, com modelos capazes de enxergar as preferências dos usuários, mesmo quando não explícitas. Utilizando Sistemas de Recomendação construídos com arquiteturas de redes neurais encontradas no estado da arte, o trabalho obteve resultados positivos e promissores em relação a geração de recomendações relevantes, mesmo com preferências dos usuários não explícitas.

Palavras-chave: Dados Implícitos. Preferências de Usuários. Sistemas de Recomendação.

ABSTRACT

Due to the e-commerce growth, the crave for mechanisms capable of personalizing user experience has been facing a continuous growth. Considering that fact, the use of Recommender Systems have been one of the biggest bets to build stronger and loyal relations between clients and the commerce itself. Getting relevant recommendations, however, is a common struggle, especially when it's not possible to get user preferences in an explicit way. Due to this demand, this study aimed to explore different recommendation models and generate recommendations with similar or better hit ratios than the recent researches, with models that are able to observe user preferences even when not explicit. By using Recommender Systems built with neural networks found on the state of the art, this study achieved promising and positive results when it comes to good recommendations, even with hidden user preferences.

Key-words: Implicit Data. Recommender Systems. User Preferences.

LISTA DE FIGURAS

1	Diagrama de recomendação baseada em conteúdo.	23
2	Estrutura da rede LSTM: Portões de entrada, saída e esquecimento e suas respectivas funções de ativação.	26
3	Funções de ativação e transformações matemáticas dos portões de entrada, saída e esquecimento de uma célula LSTM.	27
4	Diagrama metodológico da pesquisa com todas as etapas do projeto.	30
5	Dados recebidos pelo TuriCreate, colunas com identificadores de usuários e itens.	36
6	Camadas do Sistema D-Mean.	38
7	Recomendação D-Mean	40
8	Arquitetura TSTM: Transformações realizadas nas três camadas que compõe o modelo de tradução.	43
9	Arquitetura TSTM Modificada	44
10	Exemplo de recomendações erradas feitas pelo LSTM-TSTM com a base Metacritic Game Review.	47
11	Exemplo de recomendação de sucesso feita pelo LSTM-TSTM com a base MovieLens	48

LISTA DE TABELAS

1	Tabela de interação Usuário x Itens	24
2	Recomendação gerada pelo TuriCreate, sequência de itens recomendados para um usuário.	37
3	Matriz de interações, cada célula é preenchida com a quantidade de interações do usuário com o item.	39
4	Resultados obtidos com o Turicreate (Similaridade de Jaccard).	45
5	Resultados obtidos com DMean	45
6	Resultados obtidos com LSTM	46
7	Resultados obtidos com LSTM-TSTM	46

LISTA DE ABREVIATURAS

CF - *Collaborative Filter* (Filtro Colaborativo)

CNN - *Convolutional Neural Network* (Rede Neural Convolucional)

HR@K - *Hit Ratio at K* (Taxa de acerto em K)

LSTM - *Long Short Term Memory* (Memória de curto-longo prazo)

RNA - Rede Neural Artificial

RNN - *Recurrent Neural Network* (Rede Neural Recorrente)

SR - Sistema de Recomendação

TF-IDF - *Term Frequency–Inverse Document Frequency* (Frequência do Termo-Frequência Inversa do Documento)

Sumário

1	INTRODUÇÃO	18
1.1	Escopo do Trabalho	18
1.1.1	Definição do Problema	18
1.1.2	Objetivo geral	19
1.1.3	Objetivos específicos	19
1.2	Estrutura da monografia	20
2	CONCEITOS GERAIS	21
2.1	Sistemas de Recomendação	21
2.1.1	Filtros Colaborativos	21
2.1.2	Filtros Baseados em Conteúdo	22
2.1.3	Matrizes de Fatoração	23
2.2	Deep Learning	24
2.2.1	Redes Neurais Recorrentes	24
2.2.2	Redes Long-Short Term Memory	25
2.3	Métricas de Avaliação de Sistemas de Recomendação	27
3	METODOLOGIA	29
4	ESTADO DA ARTE	31
5	SISTEMAS AVALIADOS	34
5.1	Dados	34
5.1.1	Tratamento de Dados	35
5.2	TuriCreate	36
5.3	D-Mean	37
5.4	LSTM e TSTM	40
6	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	45
7	CONCLUSÕES E TRABALHOS FUTUROS	49

7.1	Conclusões	49
7.2	Trabalhos Futuros	49
	REFERÊNCIAS	50

1 INTRODUÇÃO

Relacionamentos interpessoais e entre clientes e o comércio mudaram drasticamente depois da adoção da Internet como tecnologia rotineira, no final da década de 90, e desde então, como aponta Magalhães et al. (2013), os indivíduos tornaram-se produtores e disseminadores de opiniões e conteúdos para a Internet. Consequentemente, o modo como os consumidores se relacionam com o mercado também mudou: a pessoa consumidora passou a ser mais ativa, mais crítica, mais assertiva, e logicamente, mais consciente das diversas opções de mercado que lhes estão disponíveis, principalmente quando se trata de comércio virtual (aqui referido como *e-commerce*).

No Brasil, segundo a Pesquisa Nacional por Amostra de Domicílios (PNAD), realizada pelo IBGE, e divulgada em Abril de 2020, o percentual de domicílios com acesso à Internet subiu para 79,1% em 2018 ¹, o que implica que o número de consumidores brasileiros do comércio virtual também subiu. A chegada da pandemia do vírus COVID-19, como é apontado por Kunchaparthi (2021), acabou por intensificar a popularidade de compras realizadas no *e-commerce* em 25% , só em Março de 2020. Ele afirma também que o poder do *e-commerce* não deve ser subestimado, visto que ele continua a permear no cotidiano e a apresentar oportunidades significantes para pequenos, médios e grandes negócios, assim como investidores virtuais.

Comerciantes de diversas áreas têm, então, apostado em expandir seus negócios para o *e-business*, e para se destacarem, eles precisam fornecer experiências personalizadas e interativas com os clientes. Um dos pilares para essas interatividades e personificações é o uso de sistemas de recomendação, que podem oferecer sugestões que facilitem o processo de tomada de decisões da pessoa compradora com relação ao serviço utilizado. Grandes empresas como a Netflix, Spotify e Amazon vêm se destacando com suas experiências personalizadas graças aos seus modelos de recomendação de produtos. Entretanto, a qualidade de dados com os quais essas empresas trabalham não é a realidade da grande maioria de plataformas de interações entre clientes e produtos. Dado esse fato, é apresentada, a seguir, a problemática inspiradora deste trabalho.

1.1 Escopo do Trabalho

1.1.1 Definição do Problema

Como afirma Sant’ana (2016), a satisfação do consumidor é advinda de todo o processo com a plataforma virtual: desde a página inicial, passando pela escolha do produto, efetivação da compra, entrega do produto e pós-venda. É essencial, então, para

¹Disponível em <<https://agenciabrasil.ebc.com.br/economia/noticia/2020-04/um-em-cada-quatro-brasileiros-nao-tem-acesso-internet>> Acesso em: 04 mai, 2021.

a satisfação do cliente, que o processo de escolha do produto seja, assim como as demais etapas, de qualidade.

O cliente, então, no processo de escolha, passa a ser um tomador de decisão. Segundo Chiavenato (2005, apud Lima, 2012), tal pessoa deliberadora está inserida em uma determinada situação, tem preferências pessoais, pretende alcançar um ou mais objetivos e segue cursos de ação para alcançar resultados, de forma que, toda decisão envolve uma opção.

Em sistemas *e-commerce*, é importante que essas opções sejam condizentes com as preferências pessoais e contemplem o contexto no qual a pessoa decisora (cliente) está inserida. Entretanto, os empreendedores que vêm migrando ou expandindo seus negócios para *e-business* acabam enfrentando um problema: a limitação dos dados para desenvolver SRs (Sistemas de Recomendação) que abranjam tais aspectos. Considerando que diversos empreendedores não possuem dados comumente usados em sistemas de recomendação mais personalizados, como dados demográficos ou *feedback* da experiência do usuário para com o produto adquirido, surge o problema a ser resolvido: gerar boas recomendações de itens utilizando dados que não mapeiam explicitamente informações sobre preferências do usuário, ou seja, usando preferências implícitas ao comportamento de interação do usuário.

1.1.2 Objetivo geral

O objetivo principal deste trabalho é, além de explorar modelos distintos de sistemas de recomendação, demonstrar - através de comparações entre sistemas de recomendações - que é possível gerar sugestões de itens baseadas apenas em características intrínsecas aos itens e ao comportamento de observações de usuários com os produtos.

1.1.3 Objetivos específicos

Baseando-se nos objetivos gerais deste trabalho, planeja-se:

- Explorar e escolher modelos de sistemas de recomendação considerando a proposta de gerar recomendações baseadas em comportamento de interações de usuários e não em *feedback* explícito de usuários;
- Comparar os sistemas de recomendação escolhidos no que diz respeito às suas características;
- Comparar o desempenho dos sistemas implementados com uso de métricas heurísticas utilizadas no estado da arte.

1.2 Estrutura da monografia

Esta monografia é constituída em sete capítulos, onde o primeiro (Introdução) abrange o âmbito do estudo deste trabalho, os objetivos gerais e específicos, a problemática explorada e a justificativa. O segundo (Conceitos Gerais) contempla a contextualização da pesquisa; terceiro (Metodologia) e o quarto (Estado da Arte) contemplam o processo de desenvolvimento desses modelos e as ferramentas utilizadas para tal implementação, assim como a compreensão do estado da arte da área escolhida; o quinto (Sistemas Avaliados), por sua vez, engloba a escolha de modelos de recomendação e a construção desses sistemas. O sexto capítulo (Apresentação e Análise de Resultados) traz a apresentação das bases de dados utilizadas para teste dos modelos desenvolvidos e, além disso, os resultados de desempenho dos sistemas escolhidos. O sétimo e último (Conclusões e Trabalhos Futuros) é incumbido de apresentar comentários sobre o trabalho em geral e planejamento de possíveis trabalhos futuros.

2 CONCEITOS GERAIS

2.1 Sistemas de Recomendação

Sistemas de Recomendação, como afirmado por Ricci et al. (2011), são ferramentas e técnicas que geram sugestões de itens que possam ser de interesse para um usuário. Esses itens podem ser de natureza diversa: músicas, filmes, roupas, aplicativos, etc.

Ao considerar o contexto de comércio virtual, quando os itens recomendados são relevantes para os consumidores, o sistema de recomendação pode atuar diretamente no aumento de vendas. Dessa forma, como aponta Ferro (2010), é possível enxergar que um dos maiores desafios desses sistemas é justamente a seleção de itens relevantes - ou seja, que estejam em acordo com os gostos e perfis de consumo dos usuários - para um cliente, dado um vasto catálogo de itens.

A definição formal para o desafio de gerar recomendações, ou seja, a problemática de gerar um ranking de itens de interesse pode ser compreendida como:

Seja U o conjunto de todos os usuários de um determinado sistema, e seja I o conjunto de todos os possíveis itens que podem ser recomendados como livros, filmes, restaurantes, etc. Seja f_u função utilidade que mede o quão útil é um determinado item para um determinado usuário u , i.e., $f_u : U \times I \rightarrow A$, onde A é um conjunto totalmente ordenado (e.g., inteiros não negativos ou números reais dentro de um intervalo). Então, para cada usuário $u \in U$, procura-se um item $i' \in I$ que maximiza a utilidade do usuário. Mais formalmente, isto pode ser expresso pela equação [...]:

$$\forall u \in U, i' = \operatorname{argmax}_{i \in I} f_u(u, i) \quad (1)$$

(Adomavicius and Tuzhilin, 2005)

De maneira menos formal, é possível generalizar um SR como uma sistematização de uma função de ativação f_u responsável por calcular o grau de relevância de um item i , pertencente ao conjunto de itens I , para um usuário u , pertencente a um conjunto de usuários U .

2.1.1 Filtros Colaborativos

Os sistemas de recomendação partem da definição formal, mas variam de diversas formas no que diz respeito à construção da função que calcula o grau de relevância dos itens. Um dos métodos mais famosos de SR é o filtro colaborativo (CF - *Collaborative Filtering*, em inglês). Schafer et al. (2007) definem o processo de CF como a filtragem ou avaliação de itens a partir da opinião alheia, partindo do pressuposto que humanos

compartilham opiniões com outros há centenas de anos. Basta imaginar a quantidade de vezes que indicações de pessoas levam livros a serem lidos, filmes a serem vistos, produtos a serem comprados, etc.

A aplicação de CFs surgiu em meados da década de 90, segundo Schafer et al. (2007), com a necessidade de lidar com a natureza diversa de repositórios textuais, visto que trabalhar com sistemas que representassem os produtos apenas por palavras chaves era dúbio. Clientes que procurassem por palavras chaves como, por exemplo, 'massas', poderiam encontrar conteúdo sobre densidade demográfica, ou sobre comidas, ou até mesmo conceitos físico-químicos. Dessa forma, como também aponta Schafer et al. (2007), nessa época, haviam duas soluções possíveis, esperar avanços na inteligência artificial ou trazer opiniões humanas para o processo.

Considerando essas duas estratégias, a Xerox Palo Alto Research Center desenvolveu um sistema experimental de correio chamado Tapestry, que visando lidar com a enorme quantidade de mensagens que poderiam ser enviadas aos usuários, propôs a ideia de um sistema que, segundo Goldberg et al. (1992), proveria listas de tópicos de interesse e, além disso, contaria com um filtro dessas listas, onde esse processo de filtragem seria feito pelo usuário, a partir de reações (também chamadas a anotações) de outros usuários para com os documentos listados em cada tópico. Schafer et al. (2007) aponta detalhes que descrevem o sistema Tapestry: usuários poderiam formular consultas ao sistema que combinavam: 1) informação textual básica (e.g. contém a palavra "massas"); 2) metadados semânticos (e.g. escrito pela chefe de cozinha Paolla Carossella) e 3) anotações (e.g. marcado como excelente por Joana). Foi logo depois do surgimento do Tapestry que a relevância da interação *usuário-item* passou a ser reconhecida e explorada em diversos âmbitos acadêmicos.

2.1.2 Filtros Baseados em Conteúdo

Outro tipo comum de filtragem de itens para Sistemas de Recomendação é a baseada em conteúdo, na qual o sistema, como afirma Ricci et al. (2011), tenta gerar uma lista de possíveis itens de interesse para o usuário, de acordo com itens similares que o usuário interagiu previamente. Desta forma, é possível abstrair os itens a serem recomendados em vetores de características (*embeddings*) que, em conjunto, definem o perfil do usuário.

Em outras palavras, como também afirma Ricci et al. (2011), sistemas de recomendação que fazem uso dessa metodologia, analisam *embeddings* de itens no histórico de interações de um usuário e, a partir dessas características extraídas, definem o perfil desse usuário em questão. A partir daí, o sistema passa a contrapor características do perfil do usuário com características de outros itens. O resultado dessas contraposições atrelam uma relevância a cada item em relação ao usuário. A Figura 1 abaixo ilustra

uma recomendação baseada em conteúdo.

Figura 1: Diagrama de recomendação baseada em conteúdo.



Fonte: Grimmaldi, Towards Data Science

2.1.3 Matrizes de Fatoração

Algumas estratégias de desenvolvimento de sistemas de recomendação exploram a interação entre usuários e itens, de forma que os fatores latentes, ou seja, fatores não diretamente observáveis, que caracterizam essas interações precisam ser obtidos de alguma forma. Uma maneira bastante utilizada entre estratégias como, por exemplo, filtros colaborativos, é o uso de matrizes de fatoração. Chen (2020) aponta que matrizes de fatoração são utilizadas por conseguirem gerar fatores latentes através da multiplicação de dois tipos distintos de entidades, como por exemplo, usuário e item.

O filtro colaborativo, por exemplo, como Chen (2020) também afirma, se dá através da aplicação de matrizes de fatoração para identificar relações intrínsecas entre entidades de itens e usuários. Em outras palavras, matrizes de fatoração, no caso do filtro colaborativo, servem para indicar, através das interações (que podem ser avaliações ou até mesmo quantidade de compras) de usuários com itens, como os itens seriam relevantes para os usuários.

A Tabela 1 ilustra como a relação usuário-item é obtida e traz um exemplo de como interações usuário-item são mapeadas para que fatores latentes possam ser extraídos. A

tabela é preenchida pela interação do usuário com o item, no exemplo da figura, as quantidades de vezes que o usuário interagiu com o item. Como fica claro na imagem, nem todo usuário interage com todas os itens e as "não interações" são preenchidas, na verdade, com zeros, de forma que, quanto mais itens sem interação com usuário, mais esparsas ficam a tabela e a matriz de fatoração. Essa e outras características deste tipo de abordagem são mais exploradas no Capítulo 4.

Tabela 1: Tabela de interação Usuário x Itens

	ITEM 1	ITEM 2	ITEM 3	ITEM 4
USUÁRIO 1	-	2	-	1
USUÁRIO 2	1	1	1	-
USUÁRIO 2	2	-	1	-

2.2 Deep Learning

Além dos métodos apresentados, como o uso de Matrizes de Fatoração, outras áreas vem sendo estudadas para solucionar problemas de SRs. *Deep learning* (em português, aprendizagem profunda) é um deles. É considerado um campo do aprendizado de máquina, e a característica que define esse âmbito da inteligência artificial é o fato de que sistemas de *deep learning* aprendem a representar os dados em múltiplos níveis de abstração. Ou seja, de maneira prática, modelos de aprendizagem profunda são aqueles que otimizam uma função objetivo diferenciável, com uso de um gradiente descendente estocástico (Zhang et al., 2019).

No âmbito da aprendizagem profunda, diversas técnicas computacionais podem ser desenvolvidas, dentre elas, as Redes Neurais Artificiais (RNAs), que inspiradas no sistema nervoso central animal, são capazes de aprender e reconhecer padrões. Este estudo explora o uso de aprendizagem profunda para gerar recomendações de itens. Os tópicos abaixo introduzem a família de modelos explorado nesta pesquisa.

2.2.1 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (em inglês, *Recurrent Neural Networks (RNN)*) são uma família de redes neurais que, segundo Goodfellow et al. (2016), foram feitas para processamento de dados sequenciais. São redes que, assim como as Redes Neurais Convolucionais (em inglês, *Convolutional Neural Network (CNNs)*), conseguem compartilhar parâmetros a cada iteração, entretanto, o compartilhamento de parâmetro se dá de maneira distinta ao comportamento das CNNs. Como também é apontado por Goodfellow et al. (2016), o compartilhamento de parâmetro das RNNs se dá de forma que cada membro do grupo de saídas da rede é produzido com a mesma regra de aprendizado aplicada às saídas anteriores.

No contexto de Sistemas de Recomendação, o uso de RNNs tem sido cada vez mais presente, visto que com o uso de tais tecnologias, é possível encontrar relações inerentes aos itens a serem recomendados, sob a condição de que o volume de dados utilizado no sistema seja satisfatório (Zhang et al., 2019). Em outras palavras, se a base de dados utilizada para construir o modelo de recomendação for suficientemente diversa e similar ao problema real, Redes Neurais Recorrentes podem ser uma ótima abordagem para encontrar similaridades entre itens e, conseqüentemente, gerar boas sugestões. Entretanto, mesmo que muitas abordagens de recomendação venham explorando o uso de RNNs, ainda são poucas as abordagens que usam Redes Neurais Recorrentes para realizar recomendações instantâneas, ou seja, são poucas as abordagens que utilizam RNNs e levam em consideração o contexto no qual está sendo gerada a sugestão de itens.

2.2.2 Redes Long-Short Term Memory

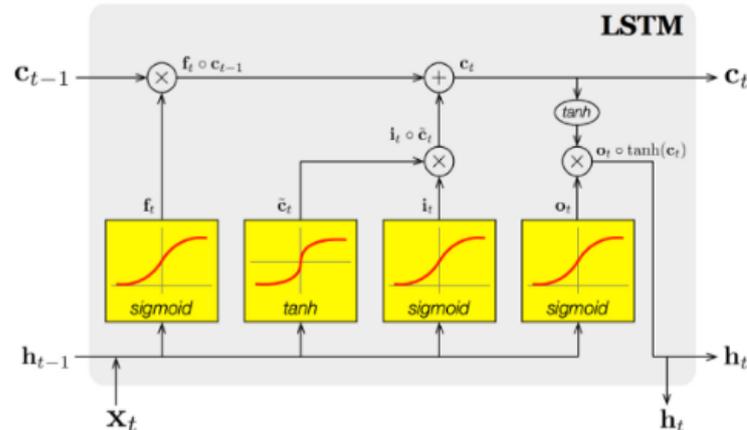
Um dos problemas enfrentados por Redes Neurais Recorrentes é a capacidade de aprender características de longo prazo - conhecidas, em inglês, como *long-term dependencies*. O problema se dá pelo fato de que o gradiente de aprendizado (algoritmo que visa de otimizar os parâmetros de aprendizado, como os pesos, encontrando os valores de mínimo das funções de aprendizado da rede), quando propagado por muitos estágios, tende a desaparecer ou em alguns casos raros, crescer mais do que o esperado, e mesmo quando a rede utilizada é uma rede estável que não permite que o gradiente exploda, os pesos ainda tendem a ficar exponencialmente menores quando iterando a longo prazo (Goodfellow et al., 2016). Logo, o uso de Redes Neurais Recorrentes simples *vanillas*, como são conhecidas popularmente, para resolução de problemas que necessitam de um aprendizado de longo prazo não é a melhor saída, visto que elas correm o risco de perder a capacidade de manter pesos antigos ao longo do aprendizado.

A Rede Neural Recorrente conhecida como Memória de Longo-Curto Prazo (em inglês, LSTM - *Long-Short Term Memory*), é uma rede capaz de aprender considerando a ordem das sequências com as quais é treinada. Em outras palavras, redes LSTM visam contornar o problema do aprendizado de *long-term dependencies*. As redes LSTM são capazes de contornar o problema com o uso de portões (ou *gates*, em inglês) que compõem as células de memória (*memory cells*, em inglês). *Memory cells* podem ser interpretadas, de maneira simplificada, como células capazes de decidir o que deve ser mantido e o que deve ser descartado durante as iterações.

A estrutura da LSTM, ilustrada na Figura 3, conta então com uma unidade chamada portão de entrada multiplicativo (*multiplicative input gate*, em inglês) que é responsável por proteger conteúdos armazenados na rede de perturbações advindas de entradas irrelevantes, e com uma unidade chamada portão de saída multiplicativo (*multiplicative output gate*, em inglês) responsável por impedir que conteúdos irrelevantes que estão

armazenados no momento sejam propagados às outras unidades da rede. (Hochreiter and Schmidhuber, 1997). Outra unidade importante na estrutura da LSTM é o portão de esquecimento (*forget gate*, em inglês) que, como o nome sugere, decide quais informações devem ser esquecidas da iteração anterior.

Figura 2: Estrutura da rede LSTM: Portões de entrada, saída e esquecimento e suas respectivas funções de ativação.



Fonte: Kang, Medium

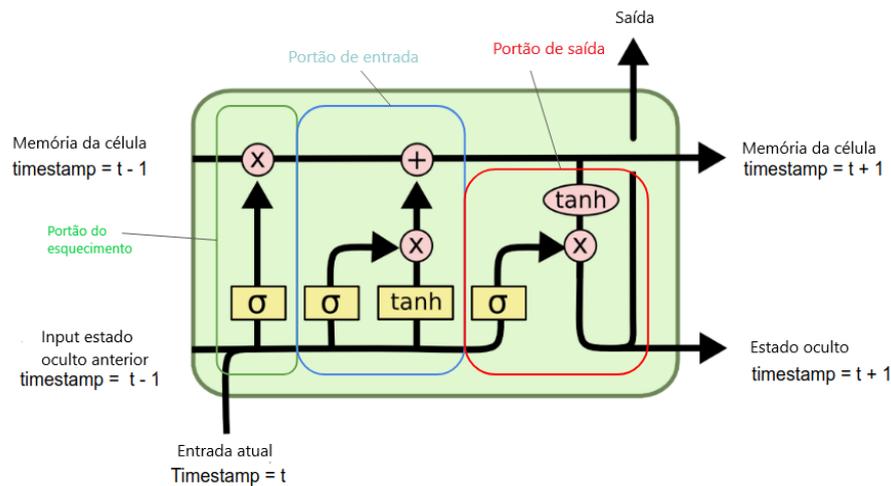
A imagem acima ilustra a estrutura das Redes LSTM, e para melhor compreensão, é interessante ressaltar que Redes Neurais se diferem de técnicas estatísticas como a regressão linear graças ao uso de funções de ativação. Ou seja, "funções de ativação são essenciais para dar capacidade representativa às redes neurais artificiais, introduzindo um componente de não linearidade" (Facure (2017)). Algumas funções de ativação conhecidas são utilizadas na estrutura da LSTM, como a sigmoide (que popularizou-se por, como Facure (2017) aponta, ser uma boa maneira de simular o comportamento binário de neurônios biológicos, já que a função assume valores entre 0 e 1) e a tangente hiperbólica (\tanh) que é uma função similar à sigmoide, exceto pelo fato de que é simétrica em relação à origem, variando então entre -1 e 1.

Na Figura 3, é possível enxergar os portões previamente citados da seguinte forma:

- Portão do esquecimento: composto pela primeira função sigmoide - da esquerda pra direita -, capaz de extrair a relevância da entrada atual, que quando submetida a um produto vetorial junto às informações atuais da célula, decide o que deve ser esquecido da memória da rede;
- Portão de entrada: composto pela tangente hiperbólica (\tanh), que extrai informações características da entrada atual, e pela segunda sigmoide, que novamente extrai a relevância da entrada atual. O produto vetorial dessas duas funções consegue adicionar na célula o que deve ser lembrado da entrada atual.

- Portão de saída: composto pela última sigmoide, que atua da mesma maneira das duas anteriores, e por uma outra tangente hiperbólica, que dessa vez atua extraindo características já salvas na memória da rede. O produto vetorial das duas funções atua como construção do que deve ser entendido como saída da célula, considerando a relevância do que acabou de entrar.

Figura 3: Funções de ativação e transformações matemáticas dos portões de entrada, saída e esquecimento de uma célula LSTM.



Fonte: Elaborada pela autora

No contexto de sistemas de recomendação, o uso de redes LSTM ainda é muito novo, mas já é bastante promissor, visto que por contornar o problema de aprender sequencialmente à longo prazo, as redes LSTM podem ser utilizadas para gerar recomendações de itens considerando o contexto da interação usuário-item, afinal, se considerarmos o contexto de compras, por exemplo, os "carrinhos" não deixam de ser sequências de itens, e sequências são a especialidade das redes LSTM.

2.3 Métricas de Avaliação de Sistemas de Recomendação

O uso de sistemas de recomendação se dá, em grande maioria, com o objetivo de facilitar a tomada de decisão de usuários no que diz respeito à interação com itens, de forma que as sugestões feitas pelos sistemas influenciem na tomada de decisão do usuário. Conseqüentemente, medir o quão bem um sistema de recomendação está se saindo não é uma tarefa trivial, visto que não se trata apenas de acurácia e erros, mas sim da capacidade do sistema de prever itens que serão, de fato, relevantes para o usuário.

Uma das técnicas utilizadas para avaliar o desempenho de um sistema de recomendação é a métrica de taxa de acerto em K (HR@K, do inglês, *Hit Ratio at K*), que essencialmente calcula a taxa de acertos (*hits*) das recomendações. Em outras palavras, um acerto é considerado se uma recomendação contém pelo menos um item com o qual o usuário interagiu. É importante salientar, como aponta Alsini et al. (2020), que essa métrica não consegue refletir a qualidade do acerto, visto que se uma recomendação top- k , com $k = 5$, tem todos os 5 itens na lista de escolhidos pelo usuário. Um *hit* é considerado, de mesma maneira, se uma recomendação de mesmo top- $k = 5$ tem apenas 1 na lista de escolhidos pelo usuário, um hit também é considerado.

Os valores da métrica HR@K são limitados, como também aponta Alsini et al. (2020), entre $[0, 1]$, onde o valor 0 indica que não houve nenhum acerto, e o valor 1 indica que todas as recomendações acertaram pelo menos 1 item. Qualquer valor entre 0 e 1 indica que algumas recomendações acertaram e outras não. Matematicamente falando, a taxa HR@K é dada pela equação 2:

$$HR@K = \frac{hits}{(hits + misses)} \quad (2)$$

Onde *hits* é o número de recomendações no qual o top- k contém pelo menos 1 acerto, e *misses* é o número de recomendações no qual o top- k não contém nenhum acerto.

No intuito de atingir os objetivos deste estudo, foram utilizadas técnicas apresentadas neste capítulo, tais como o uso de redes neurais recorrentes LSTM e filtros baseados em conteúdo no desenvolvimento dos sistemas avaliados neste trabalho, como também a métrica HR@K para avaliação de desempenho dos mesmos. O capítulo 4 (Estado da Arte) traz consigo uma exploração do estado da arte de sistemas de recomendação, onde técnicas como as apresentadas têm sido fortemente aplicadas em diversas pesquisas, e o capítulo 5, por sua vez, está encarregado de expor mais detalhadamente como as estratégias de desenvolvimento e avaliação exibidas neste capítulo foram utilizadas para o alcance dos objetivos apresentados no capítulo 1.

3 METODOLOGIA

Este capítulo traz consigo a estrutura do trabalho desenvolvido durante a trajetória deste projeto e explana as características do tipo de pesquisa. Antes de abordar a disposição dos passos do trabalho, é importante compreender o tipo de pesquisa realizada. Sobre tipos de pesquisa, é possível afirmar que:

Diferentemente da pesquisa qualitativa, os resultados da pesquisa quantitativa podem ser quantificados. Como as amostras geralmente são grandes e consideradas representativas da população, os resultados são tomados como se constituíssem um retrato real de toda a população alvo da pesquisa. A pesquisa quantitativa se centra na objetividade. Influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno, as relações entre variáveis, etc. A utilização conjunta da pesquisa qualitativa e quantitativa permite recolher mais informações do que se poderia conseguir isoladamente (Fonseca, 2002).

Considerando que o projeto teve o objetivo de comparar modelos de sistemas de recomendação treinados com dados implícitos, e tem seus resultados avaliados de forma quantitativa, em cima de uma população dos dados utilizados, é seguro dizer que o trabalho é de caráter quantitativo. Entretanto fez-se necessária, inicialmente, a coleta de informações sobre o objeto de estudo - sistemas de recomendação - e esta etapa se deu através de uma pesquisa exploratória, com um levantamento bibliográfico e uma análise de exemplos. Em outras palavras, a primeira etapa do trabalho se deu através de um estudo capaz de prover uma familiaridade maior com o problema, e a segunda etapa do trabalho se deu através de uma comparação quantitativa dos objetos de estudo.

No que diz respeito à problemática da pesquisa, para conseguir sistemas capazes de respondê-la, foram seguidos alguns passos. Eles estão apontados a seguir:

1. Pesquisa Exploratória

Levantamento bibliográfico no contexto de sistemas de recomendação, e análise de exemplos de sistemas de recomendação, ambos visando maior compreensão do assunto.

2. Escolha dos dados

Procura e escolha de bases de dados disponíveis que pudessem ser utilizadas nos testes dos Sistemas de Recomendação escolhidos para solucionar o problema, como também pudessem ser utilizadas para comparação de desempenho com o estado da arte.

3. Tratamento das bases de dados

Pré-processamento das bases de dados escolhidas, para que elas pudessem ser utilizadas em todos os SRs explorados no trabalho. Limpeza da base, agrupamento de interações por usuários e outras etapas, apresentadas no Capítulo 5, foram realizadas neste ponto do trabalho.

4. Escolha e implementação dos sistemas

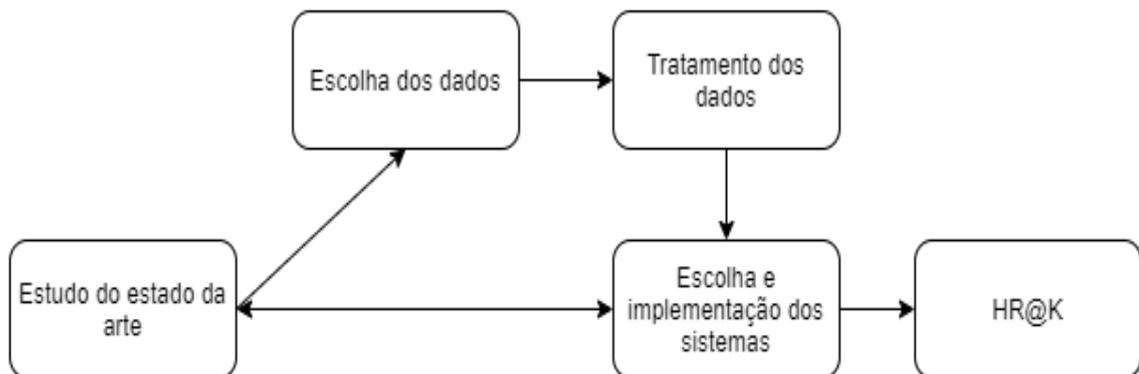
Com base no estudo do estado da arte, foram escolhidos uma ferramenta de recomendação já desenvolvida; um SR baseado em sistemas de tradução construídos com Redes Neurais Recorrentes - a ser implementado; e um combinado - também a ser implementado - de ferramentas de recomendação e filtros de personalização. Mais informações sobre os três SR escolhidos, e implementados, estão presentes no próximo capítulo.

5. Avaliação dos sistemas

Na fase final, foi utilizada uma técnica comum na avaliação de Sistemas de Recomendação: 'Taxa de acerto em K', *Hit Ratio at K* (HR@K) em inglês, e segundo Huang (2018), caracteriza a fração de acertos do sistema, no que diz respeito ao top- k produtos recomendados, e.g., seja k igual a 5, e seja o número de recomendações feitas igual a 10, serão consideradas os top 5 itens para cada um dos 10 usuários; se dessas 10 recomendações, 3 usuários interagiram com pelo menos um dos top 5 itens recomendados, um acerto (*hit*) será contabilizado, de forma que o HR@5 dessas recomendações será de 0,3. Além disso, foram comparados os HR@Ks de cada sistema implementado com taxas encontradas no estado da arte.

Antes de prosseguir, é importante ressaltar que diversas vezes essas etapas foram revisitadas durante o ciclo de desenvolvimento da pesquisa. Um diagrama que ilustra uma abstração do processo de desenvolvimento da pesquisa é encontrado na Figura 4.

Figura 4: Diagrama metodológico da pesquisa com todas as etapas do projeto.



Fonte: Elaborada pela autora

4 ESTADO DA ARTE

A literatura atual explora diversas abordagens de sistemas de recomendação, considerando contextos como a recomendação de itens através do uso de dados explícitos, as avaliações realizadas por usuários, e também a recomendação de itens com o uso de dados implícitos. Nesse âmbito, algumas abordagens foram estudadas e serão apresentadas neste capítulo.

Grandes empresas como Apple Inc. e Netflix vêm apostando em desenvolvimentos experimentais de sistemas de recomendação. A Apple, por exemplo, na Apple Worldwide Developers Conference de 2018 ², apresentou um conjunto de ferramentas de aprendizagem de máquina em código aberto, TuriCreate³, que contava com um pacote destinado a sistemas de recomendação. Já a Netflix vem explorando descobertas acadêmicas com o Netflix Research, e com desafios, como o primogênito da Netflix no âmbito de sistemas de recomendação: Netflix Prize ⁴, um desafio que encorajou estudiosos e programadores a tentarem achar a melhor solução de filtragem colaborativa para prever as classificações de usuários para os filmes da plataforma. Nos parágrafos abaixo, são apresentadas algumas das abordagens encontradas na literatura.

A documentação do TuriCreate destinada aos sistemas de recomendação inclui três tipos de cenários possíveis para usar o TuriCreate como modelo base de recomendação: a partir de dados explícitos, como avaliações de itens feitas por usuários; a partir de dados implícitos, onde as recomendações seriam oriundas da similaridade entre itens com os quais usuários interagiram previamente; e por fim, a partir de dados intrínsecos aos itens, e não à interação usuário-item.

Abordagens como a proposta pelo pacote de sistemas de recomendação com dados implícitos do TuriCreate usualmente recorrem a métricas de similaridade entre itens para definir um ranking de relevância de itens para os usuários. A Apple, por exemplo, neste *toolkit*, disponibiliza três coeficientes de similaridades bastante utilizados no estado da arte para o cálculo da recomendação para filtros colaborativos, Ricci et al. (2011) apresenta algumas utilizadas no TuriCreate:

1. Índice de *Jaccard* - obtido da divisão entre interseção e união de um par de listas (vetores) de itens, por exemplo, considerando dois históricos de compra c_1 e c_2 , e calculando a interseção I_c entre os itens de cada um dos dois históricos, assim como a união U_c de c_1 e c_2 , é possível obter o índice de *Jaccard* através da divisão $\frac{I_c}{U_c}$

²Disponível em <<https://developer.apple.com/videos/play/wwdc2018/712/>> Acesso em: 04 mai, 2021.

³Disponível em: <<https://apple.github.io/turicreate/docs/userguide/recommender/>> Acesso em: 07 mai, 2021.

⁴Disponível em: <<https://www.netflixprize.com/>> Acesso em: 08 maio, 2021.

2. Similaridade de cosseno - obtido através da distância dos cossenos entre dois vetores. Podemos considerar os mesmos dois históricos c_1 e c_2 do item anterior, agora com avaliações dos usuários sobre os itens. Se colocados em um plano, é possível enxergar que quando o cosseno entre os vetores for próximo de 1, o ângulo entre os vetores será próximo de 0, indicando que os vetores são muito próximos e, conseqüentemente, similares.
3. Coeficiente de *Pearson* - coeficiente que indica a relação estatística entre duas variáveis, obtido através do cálculo da correlação entre elas, por exemplo, considerando novamente os dois históricos de compras c_1 e c_2 citados previamente, e seus respectivos vetores de avaliações, o coeficiente de correlação entre os dois vetores indicará o quanto o usuário u_1 , dono do histórico c_1 , concorda com o usuário u_2 , dono do histórico c_2 , perante suas avaliações.

Metodologias semelhantes às apresentadas pelo TuriCreate também foram exploradas nos desafios propostos pela Netflix, e em diversas outras fontes de estudo. Entretanto, é interessante ressaltar que para realizar cálculos de similaridades para os filtros colaborativos, é necessário obter os vetores de interação usuário-item, visto que os cálculos são baseados em vetores de interação - independente do conteúdo dos vetores ser explícito ou implícito. Para isso, são utilizadas as matrizes de fatoração previamente explanadas. Um dos maiores problemas de recomendação com uso de matrizes de fatoração envolve o fato de que os sistemas baseiam-se sempre nas preferências e observações estáticas dos usuários, acarretando a necessidade frequente de recálculo do ranking de relevância de itens para cada nova interação de um usuário com algum produto, ou simplesmente sugestões estáticas e desatualizadas.

Visando contornar o problema, diversos pesquisadores tem apostado em sistemas sequenciais de recomendação, que levam em consideração não só as observações estáticas, mas também contextos que atribuem uma certa dependência latente entre as interações no histórico de interações de cada usuário. Rendle et al. (2010) trouxe o conceito de um sistema sequencial de recomendação baseado em cadeias de Markov que, segundo Souza (2013), são processos estocásticos - coleção de variáveis aleatórias capazes de descrever o comportamento de um processo no decorrer do tempo - onde a evolução do processo se dá de um estado para o outro, ou seja, dependendo apenas do seu último estado. Em outras palavras, esse tipo de sistema parte do pressuposto que a interação com próximo item de um usuário será baseada na última ou nas últimas interações realizadas por ele. Entretanto nem sempre as últimas interações definem as próximas, mas sim o comportamento de um usuário numa janela de tempo maior.

Para abranger essas janelas maiores de tempo, quando se tem uma quantidade suficiente de dados, podem ser empregadas técnicas de aprendizado profundo, tais como

Redes Neurais Recorrentes. Neste aspecto, Kang and McAuley (2018) apresentaram o SASRec, um sistema de sugestões de produtos que consegue capturar o contexto de interações usuário-item a longo prazo - utilizando Redes Neurais Recorrentes, e a curto prazo - utilizando mecanismos *Attention* que, se descritos de maneira simples, são mecanismos capazes de levar em consideração estados intermediários comumente desconsiderados em redes que trabalham com codificação e decodificação de sequências. Esses estados intermediários, por sua vez, conseguem captar informações contextuais presentes em sequências de dados longas, como, por exemplo, textos ou histórico de compras.

Outra abordagem com mecanismos *Attention* foi apresentada por Zhang et al. (2019): o AttRec. O modelo consiste de um módulo *Self-Attention* - variação do mecanismo *Attention* - para, assim como o do SASRec, compreender interações de curto prazo, e de uma métrica colaborativa de aprendizado para compreender preferências de longo prazo do usuário. Compreendendo o contexto literário no qual o estudo de sistemas de recomendação se encontra atualmente, são apresentados, no próximo capítulo, os modelos escolhidos para implementação e realização de testes que visaram alcançar bons resultados de recomendação com dados implícitos, ou seja, dados latentes das preferências dos usuários.

5 SISTEMAS AVALIADOS

Conhecendo o que tem sido estudado na literatura, no âmbito de sistemas de recomendação, alguns deles foram escolhidos para gerar recomendações com dados implícitos, visando encontrar pelo menos uma alternativa com boas taxas HR@K. Neste trabalho, foram utilizadas duas bases de dados e quatro abordagens de recomendação. Os próximos tópicos explanam mais informações sobre as bases e o tratamento, assim como detalham as abordagens escolhidas.

5.1 Dados

Para realizar os testes previstos, foram necessários dados implícitos, visto que o objetivo deste trabalho foi traçado em torno do uso de dados implícitos em sistemas de recomendação. Duas bases de dados foram utilizadas para realização dos testes:

- Metacritic Game Review⁵: Base de dados de avaliação de jogos. A base continha inicialmente 284000 instâncias de interação, 133666 usuários e 3438 itens.
- MovieLens ⁶: Base de dados pública de avaliações de filmes. A base continha inicialmente 20000000 instâncias de interação, 138000 usuários e 27000 itens.

A escolha da base MovieLens se deu principalmente pois ela foi utilizada também no modelo TSTM proposto por Sun and Qian (2020), modelo que será utilizado para comparação de resultados. A base Metacritic, entretanto, foi selecionada para substituir a outra base utilizada por Sun and Qian (2020), visto que a segunda base utilizada pelos autores contava com dados não muito relevantes no contexto atual (dados de um aplicativo de *check-in* de locais, o aplicativo foi descontinuado e, além disso, essa categoria de aplicativos não faz mais parte do cotidiano dos brasileiros. Como o objetivo da pesquisa é trabalhar com preferências latentes de usuários, escondidas nas relações de histórico de interações usuário-item, ou seja, dados implícitos, as avaliações dos filmes e jogos, nas bases, foram transformadas em quantidades de visualizações do item avaliado. Tal transformação foi feita considerando que uma pessoa que avaliou o item, numa escala de 1 a 5, com a nota máxima (5) é muito mais propícia a interagir novamente como ele do que alguém que o avaliou com nota mínima (1). Em outras palavras, mapear a pontuação dada ao item como quantidade de interações do usuário com o filme não é uma abstração ilógica. Afim de obter os dados estruturados da maneira propícia para executar o treinamento dos modelos, foi realizado um tratamento apresentado no próximo tópico.

⁵Disponível em <<https://www.kaggle.com/dahlia25/metacritic-video-game-comments>> Acesso em: 02 jul, 2021.

⁶Disponível em <<https://grouplens.org/datasets/movielens/>> Acesso em: 02 mar, 2021.

A coluna de quantidade, simulada pela avaliação, só foi utilizada nos sistemas de filtro colaborativo que necessitavam da quantidade de interações com o item.

5.1.1 Tratamento de Dados

Essa etapa contou com a retirada de valores nulos e balanceamento das interações, visto que alguns dos sistemas - como será apontado no próximo capítulo - utilizam sequências de interações de tamanhos pré-definidos; transformação de identificadores de produtos e usuários em *tokens* (no caso, números inteiros), e agrupamento de itens por usuários e/ou compras. No caso dos modelos com arquiteturas de redes neurais, as sequências de itens e/ou categorias também passaram por uma camada de transformação em vetores de características *embeddings*, permitindo que as redes enxerguem os vetores de itens através de representações contínuas e distribuídas. Após o tratamento dos dados, foi escolhida uma amostra aleatória de cada base - nem todos os dados foram utilizados por limitações de *hardware* no treinamento com alguns dos modelos, e para fins de avaliações justas, as limitações foram aplicadas no uso das bases em todos os modelos. As bases finais continham as seguintes características:

- Metacritic Game Review: 283982 instâncias de interações, 133664 usuários e 3438 itens.
- MovieLens: 89700 instâncias de interações, 4485 usuários e 4991 itens.

No que diz respeito à separação das bases em treino e teste, foram utilizadas duas abordagens, uma para os modelos de recomendação estática (focadas nos usuários já computados), e outra focada nos modelos capazes de gerar recomendações para novos usuários:

1. Foco em usuários já computados: 100% dos usuários foram utilizados para treino, logo, o treino contou com o número de sequências únicas de itens igual ao número total de usuários únicos da base. Para cada entrada, ou seja, para cada usuário, a sequência de treino contava com 80% dos itens, enquanto que a de teste possuía os 20% restantes. Em outras palavras, para cada usuário, 80% de suas interações foram utilizadas como entrada, e o restante utilizadas como alvo.
2. Foco em novos usuários: 80% dos usuários foram utilizados para treino, logo, o treino teve 80% das sequências de interações geradas por cada usuário único da base. Os 20% de usuários restantes foram utilizados para gerar as sequências de interações utilizadas para teste. Nesse caso, o teste foi realizado com metade da sequência como entrada e a outra metade como alvo.

5.2 TuriCreate

Como o objetivo do trabalho era obter boas recomendações sem informações explícitas, como por exemplo, avaliações de itens, sendo o fator determinante da recomendação, foram utilizadas apenas informações sobre se a interação usuário-item existiu ou não. Para tanto, o fator de similaridade escolhido foi o índice de Jaccard, previamente apresentado no Capítulo 4, visto que no modelo disponibilizado pelo TuriCreate, o índice é calculado, como é apontado na documentação, da seguinte forma:

$$S(i, j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \quad (3)$$

Onde U_i é o conjunto de usuários que interagiu com o item i , enquanto U_j é o conjunto de usuários que interagiu com o item j . O motivo da escolha do índice de Jaccard como fator de similaridade se deu através do fato de que a similaridade é calculada em cima da união e da interseção das interações dos usuários, ou seja, considerando apenas se um usuário interagiu ou não com algum item, e nada além disso.

Para gerar os cálculos de similaridade e realizar as recomendações, o modelo, quando construído com o índice de Jaccard, precisa de duas colunas, advindas dos dados, como alvos para construção da recomendação: usuários e itens. A Figura 6 ilustra um exemplo de como os dados são passados para o modelo.

Figura 5: Dados recebidos pelo TuriCreate, colunas com identificadores de usuários e itens.

	userId	input_prod
0	36	145
0	36	163
0	36	196
0	36	485
0	36	555

Fonte: Elaborada pela autora

Utilizando o modelo de similaridade de item que, como dito anteriormente, gera recomendações a partir das interações em comum de usuários, é possível obter uma tabela de recomendação de itens para cada usuário. Essa tabela conta com quatro colunas: uma identificadora de usuários, uma identificadora de itens, uma com a pontuação de relevância

do item para o usuário e, por fim, uma com a posição do item no ranking de relevância para o usuário. A partir dessa tabela, é possível construir uma lista de recomendações para cada usuário, e calcular a taxa de acerto das recomendações, e conseqüentemente avaliar a qualidade das recomendações, como será apontado no capítulo 6. Um exemplo de recomendações é ilustrado na Tabela 2

Tabela 2: Recomendação gerada pelo TuriCreate, seqüência de itens recomendados para um usuário.

Id do Usuário	Itens Recomendados								
-gbrl	442	1106	143	699	40	174	841	1219	86
00-z3N	329	171	923	1381	632	262	611	457	557
Sarmad	323	510	212	835	105	805	159	395	937
Tuld000	16	391	311	660	53	17	27	21	395
Liteyear	565	616	622	378	539	564	451	333	216

5.3 D-Mean

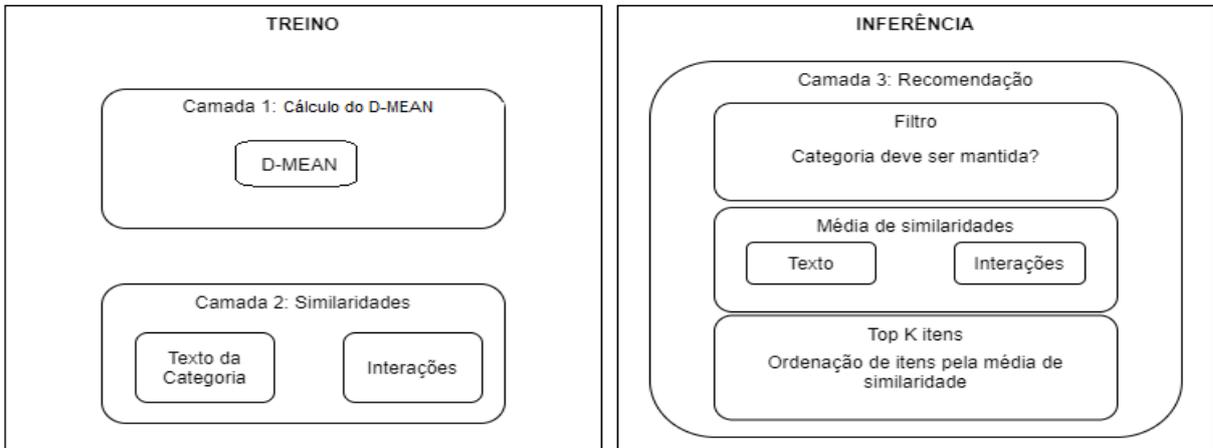
O modelo do TuriCreate, como visto anteriormente, gera recomendações para os usuários, e não para itens. Visando gerar boas recomendações, mas com uma abordagem focada nos itens, ou seja, focada em apontar itens semelhantes aos itens que estão sendo escolhidos por um usuário, foi desenvolvido neste trabalho um novo sistema, apelidado de *D-Mean*. Este sistema baseia-se na perspectiva de obter recomendações considerando as interações correntes dos usuários com os itens e as categorias desses itens, em outras palavras, funciona como um tipo de 'adicione ao carrinho' que vemos em sites como Amazon⁷ ou Rappi⁸.

Este sistema, se abstraído, consiste em três camadas que, quando combinadas, geram as recomendações de itens. Estas camadas são ilustradas na figura 6.

⁷Disponível em <<https://www.amazon.com.br/>> Acesso em: 16 mai, 2021.

⁸Disponível em <<https://www.rappi.com.br/>> Acesso em: 16 mai, 2021.

Figura 6: Camadas do Sistema D-Mean.



Fonte: Elaborada pela autora

A primeira camada consiste no cálculo de uma média que serve de limiar para indicar quando um produto está sendo "adicionado ao carrinho", ou seja, sendo escolhido, se outros produtos da mesma categoria devem ser recomendados ou não. Essa média foi chamada de 'D-Mean', visto que sua essência foi obtida da heurística apontada por Brovman (2019), onde ele calcula d como sendo a média do número de compras por usuário para cada categoria. O valor d , então, pode ser expresso pela equação 4:

$$d = \frac{\text{Compras da categoria por user}}{\text{Total de Users}} \quad (4)$$

Com d calculado, ele define um limiar para o valor de d que indica se, para uma categoria c , existe uma chance alta de mais de um item ser escolhido. Em outras palavras, o que Brovman (2019) aponta é: dado um produto qualquer, como, por exemplo, um *smartphone*, que pertence a categoria 'celulares', há uma chance d de que o usuário queira interagir com outro produto da mesma categoria, no caso, outro *smartphone* e, caso essa chance seja alta, é interessante manter todos os itens da categoria 'celulares' disponíveis para recomendação, mas caso a chance seja baixa, é mais interessante remover tais itens e abrir espaço pra outros itens.

A segunda camada consiste no cálculo de similaridade de cossenos entre as categorias, são realizados cálculos de similaridade considerando dois fatores:

1. Similaridade entre as palavras;
2. Similaridade no padrão de interação.

Para realizar os cálculos de similaridade entre as palavras das categorias dos itens (e.g., o filme 'A Viagem de Chihiro' tem como categoria 'Animação'), uma matriz de

importância das palavras de cada categoria é criada, fazendo uso de uma medida estatística conhecida como *TF-IDF*, do inglês *Term Frequency - Inverse Document Frequency* (Frequência do Termo-Frequência Inversa do Documento). Essa medida, como aponta Yun-tao et al. (2005), é comumente utilizada para medir a importância de uma palavra em um texto, dado o quão única/singular ela é. O cálculo do peso de importância se dá pela multiplicação dos valores *TF*, que é proporcional à frequência de um termo (palavra) no texto, e *IDF*, que é o valor inverso da frequência no texto. O interessante dessa função *TF-IDF*, como aponta Sebastiani (2002), é que ela encapsula duas intuições: (i) quanto mais uma palavra aparece em um único texto, mais representativa do texto ela é, e (ii) quanto mais uma palavra aparece em vários textos, menos representativa ela é.

Uma vez que a matriz de importância das palavras das categorias tenha sido construída, o sistema então calcula a similaridade de cossenos entre a matriz e ela mesma, em outras palavras, calcula a similaridade conferindo todos os pares de palavras, gerando um score de similaridade entre cada palavra que compõe e representa uma categoria.

A segunda similaridade, que abrange o padrão de interações dos usuários para com os itens, é calculada em cima da quantidade de interações que cada usuário tem com cada produto. Em outras palavras, a frequência de interações que cada item recebe dos usuários torna-se fator de peso para apontar possíveis semelhanças. Para tanto, uma matriz de interação é construída, apontando a quantidade de vezes que cada usuário interagiu com cada item, e ela fica da forma apontada na Tabela 3. Uma vez que essa matriz é construída, o sistema então calcula a similaridade de cosseno entre todos os pares da matriz.

Tabela 3: Matriz de interações, cada célula é preenchida com a quantidade de interações do usuário com o item.

userId	36	37	39 52	166	177	24	250
moviesId							
1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	5	0	0	5

A terceira e última camada, é a que gera as recomendações. Para tanto, recebe um código identificador de item, checa se a categoria do item deve ser mantida ou não nas recomendações, a partir da *D-Mean* calculada para a categoria do item, depois resgata os produtos mais próximos dentro das categorias similares - obtidas através da similaridade entre as palavras que constituem a categoria do item, e os produtos mais próximos em relação ao padrão de interação dos usuários. Depois pega o valor de similaridade de cada produto sugerido pelos rankings de similaridade de categoria e de padrão de compra, e

contabiliza a média entre as duas similaridades. Por fim, ordena os itens sugeridos pela média das duas similaridades. A figura 7 ilustra um exemplo de recomendação feita para um item.

Figura 7: Recomendação D-Mean

```
Sugestões por similaridade de interações:  
[(0.19611613513818402, 2052), (0.11094003924504584, 730), (0.04828045495852676, 737), (0.0, 641)]  
  
Sugestões por similaridade de categorias:  
[(1.0, 730), (0.24681375187369628, 2052), (0.0818807122729959, 737), (0.0818807122729959, 641)]  
  
Produtos recomendados:  
[(730, 0.555470019622523), (2052, 0.22146494350594015), (737, 0.06508058361576133), (641, 0.04094035613649795)]
```

Fonte: Elaborada pela autora

Quando a entrada do sistema é uma lista de itens, a recomendação é feita da seguinte forma: para cada elemento da lista, é gerada uma lista de sugestões, seguindo os passos descritos acima. Depois as sugestões de cada produto são concatenadas numa grande lista, que é ordenada pelo valor de similaridade de cada item sugerido, gerando assim um *ranking* de sugestões para cada produto da lista de entrada.

5.4 LSTM e TSTM

O terceiro modelo escolhido foi uma versão simples da rede LSTM, e o motivo da escolha de testar o desempenho de uma rede LSTM foi originado da vontade de entender a capacidade da rede de gerar recomendações por si só, projetando adicioná-la como mudança no modelo *Tripled Seq2seq Translation Model* (TSTM), proposto por Sun and Qian (2020).

As redes LSTM, como apresentado no Capítulo 2, são capazes de compreender relações sequenciais dentro de um curto período de tempo, considerando também os padrões de longo prazo. Esta capacidade de abstração das redes LSTM chama atenção no âmbito do estudo de sistemas de recomendação, já que é possível supor que uma lista de interações de um usuário como, por exemplo, uma lista de compras em um supermercado, não deixa de ser uma sequência de itens, especialidade das redes *Long-Short Term Memory*. No caso de uma lista de compras de mercado, por exemplo, se o problema for descobrir qual item o usuário irá comprar na interação atual (gerar uma recomendação relevante), e for considerado que das últimas 10 compras de um usuário, 8 contém a sequência 'iogurte, granola, mel', mesmo a última sequência comprada pelo usuário sendo 'iogurte, granola, pasta de amendoim', é muito mais provável que este usuário escolha 'mel', e não 'pasta de amendoim'. As redes LSTM, por considerarem as sequências de curto e longo prazo, conseguem compreender bem esses padrões e este fato foi outro fator

decisivo na escolha da não só implementação da rede LSTM, como agregação da rede no modelo TSTM original.

O quarto e último sistema escolhido foi uma versão modificada do *Tripled Seq2seq Translation Model* (TSTM), um modelo proposto por Sun and Qian (2020), que visa trazer o contexto do item como fator influenciador na decisão do usuário. Sun and Qian (2020) apontam que os sistemas que consideram o contexto na hora de gerar recomendações, usualmente não levam em conta a dependência sequencial das interações que um usuário realiza, por exemplo, quando um usuário compra um celular *smartphone*, é bastante provável que ele queira comprar um acessório (uma capinha, por exemplo) logo depois. Logo, é uma abordagem interessante, visto que o comportamento de um usuário, como Sun and Qian (2020) afirmam, pode ser extremamente influenciado pelas ações que ele toma no momento corrente. Nesta seção, a essência do TSTM será explicada, assim como também a modificação realizada para este trabalho.

Para construir o sistema TSTM, Sun and Qian (2020) inspiraram-se nos avanços recentes no âmbito da tradução automática neural (em inglês, *neural machine translation* - NMT), que visa traduzir sentenças (sequências de palavras) de um idioma para outro idioma, considerando não só o comportamento sequencial das palavras, como também relações semânticas. Os autores também definem a estrutura de um modelo NMT como composta por um codificador que lê as sentenças de entrada; um decodificador que gera a sentença traduzida; e uma "ponte" que modela as relações entre as sequências. Os codificadores e decodificadores são usualmente construídos, como indicado por Sun and Qian (2020), com redes neurais recorrentes, e a ponte com mecanismos *Attention*.

Inspirado nos modelos NMT, Sun and Qian (2020) utilizam uma arquitetura de tradução de sequências, a *seq2seq*, que permite que uma sequência de entrada seja convertida para um domínio diferente do inicial, por exemplo, se uma tradução de idiomas for considerada, podemos ter uma abordagem *seq2seq* para realizar a seguinte tradução: "Eu gosto de abacaxi" → "I like pineapple". É resgatado, então, o mecanismo *seq2seq* para conseguir codificar relações do tipo item-categoria e categoria-item, como uma espécie de tradução. Entretanto, uma diferença grande existe entre tradução textual e tradução item-categoria-item, visto que numa tradução textual, cada palavra de entrada tem uma palavra alvo correspondente, enquanto que numa tradução categoria-item, cada categoria de entrada pode conter mais de um item alvo. Para contornar esta diferença, Sun and Qian (2020) propõem o uso de um codificador automático variacional (em inglês, *variational auto-encoder* - VAE), que se trata de um modelo gerador (capaz de gerar novas instancias de dados) proposto por Kingma and Welling (2013) para filtrar informações e modelar relações subsidiárias do tipo item-categoria, de forma que as categorias dos itens sejam consideradas como o contexto de uma sequência de entrada. O uso do VAE é interessante para capturar as relações intrínsecas entre itens e categorias, visto que ele possui

um propriedade capaz de regularizar a distribuições de codificações feitas, de forma que seu espaço latente seja bom o suficiente para gerar novos dados a partir dos codificados (Rocca, 2019).

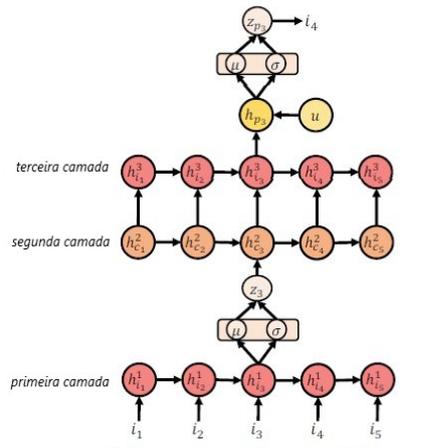
Para formular o problema do sistema, é assumido um conjunto de usuários U e um conjunto de itens I . Para cada usuário $u \in U$, uma sequência de interações é ordenada por tempo, de forma que a sequência de um usuário é definida por $s_{ui} = \{i_1^u, i_2^u, \dots, i_t^u\}$, onde i_t^u denota o item com o qual o usuário u interagiu no intervalo de tempo t . É considerado também que para cada item i , escolhido pelo usuário u , no intervalo t , possui informações contextuais adicionais como, por exemplo, as categorias - que são utilizadas como contexto no problema do sistema, fazendo com a sequência $s_{ui} = \{i_1^u, i_2^u, \dots, i_t^u\}$ de um usuário possua sempre uma sequência de categorias correspondente $s_{uc} = \{c_1^u, c_2^u, \dots, c_t^u\}$ (Sun and Qian, 2020).

Uma vez que são compreendidos os sequenciamentos de interações usuário-item e usuário-categoria, é possível compreender a arquitetura do sistema TSTM. O sistema possui dois estágios: (i) tradução de uma sequência de itens para uma sequência de categorias; (ii) tradução de uma sequência de categorias para uma sequência de itens. Sun and Qian (2020) acreditam que sequências de itens podem ajudar a prever categorias mais precisamente, e uma boa sequência de categoria gerada pode melhorar a predição de itens. Para realizar a tradução item-categoria, é utilizado um codificador automático variacional, que infere, em cada intervalo de tempo, uma variável latente obtida de uma amostragem da codificação de um item da mesma categoria do item do intervalo de tempo anterior. O uso do VAE e da variável latente se faz necessário considerando que a aplicação direta de um modelo NMT para a tarefa de tradução não seria trivial, afinal, uma categoria engloba muito mais do que um item. A diagramação da arquitetura do TSTM apresentado por Sun and Qian (2020) é ilustrado na figura 8.

A primeira camada apresentada no diagrama se trata de uma rede neural recorrente que é alimentada pela sequência de itens $s_{ui} = \{i_1^u, i_2^u, \dots, i_t^u\}$ em forma de vetores numéricos (conhecidos popularmente pelo termo em inglês '*embeddings*'). A RNN então gera os estados ocultos relacionados aos itens de entrada. Entre a primeira e segunda camada, é inserido um VAE para gerar uma amostragem dos estados ocultos dos itens em um espaço latente.

A segunda camada então gera predições de categorias ao incorporar a variável latente obtida com o VAE à sequência de categorias $s_{uc} = \{c_1^u, c_2^u, \dots, c_t^u\}$ vetorizadas, gerando os estados ocultos das categorias. A terceira camada, por sua vez, utiliza os estados ocultos obtidos segunda camada e a sequência de itens para compreender a relação subsidiária entre as categorias e os itens. Logo após a terceira camada, é inserido um vetor de preferências de usuário, visto que os autores consideram que preferências de longo prazo podem ser importantes para definir a próxima escolha de um usuário. O último passo do

Figura 8: Arquitetura TSTM: Transformações realizadas nas três camadas que compõe o modelo de tradução.

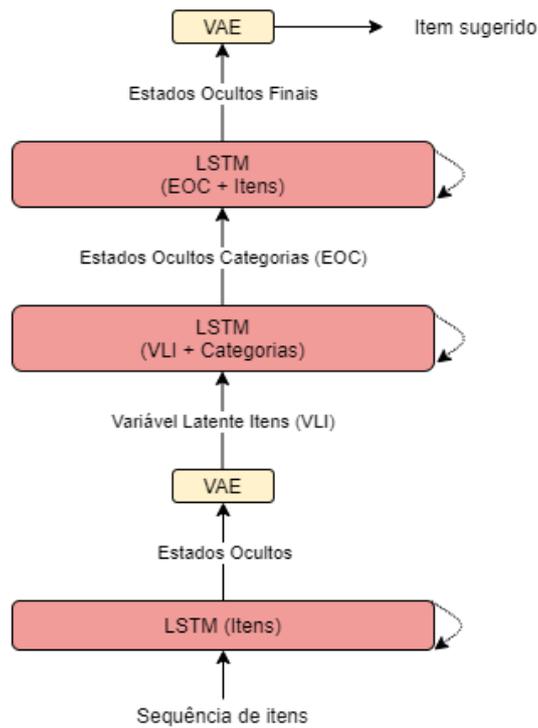


Fonte: Sun and Qian (2020)

sistema é gerar uma nova amostragem dos estados ocultos finais, visto que ao utilizar o VAE nesta etapa, o modelo ganha robustez, graças aos ruídos que são adicionados com a amostragem (Sun and Qian, 2020).

Neste trabalho, entretanto, o TSTM não foi utilizado na sua arquitetura pura, que foi apresentada previamente. Pensando na proposta de não utilizar avaliações explícitas do usuário, a sub-camada que adiciona o vetor de preferências de um usuário foi retirada, visto que para obter informações sobre quais itens eram favoritos do usuário, informações explícitas precisariam ser utilizadas. A outra modificação utilizada foi a troca da RNN padrão, conhecida popularmente como *vanilla*, por uma rede neural recorrente do tipo LSTM. Um diagrama abstraído da arquitetura modificada do TSTM que foi utilizada neste trabalho é ilustrado na Figura 9.

Figura 9: Arquitetura TSTM Modificada



Fonte: Elaborado pela autora

Os modelos escolhidos para avaliação neste trabalho são, então, o recomendador por similaridade de itens do TuriCreate; o recomendador utilizando filtro D-Mean, o recomendador utilizando uma rede LSTM simples e, por fim, o sistema TSTM modificado. O próximo capítulo apresenta o desempenho dos três modelos nas duas bases apresentadas no início desta seção.

6 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Os três sistemas de recomendação apresentados na seção anterior foram testados nas bases Metacritic Game Review e MovieLens, como previamente dito. O desempenho dos sistemas foi medido com o uso da métrica 'taxa de acerto' para *top k* itens recomendados HR@K com valores assumidos para o *top k* sendo $k = 5$, $k = 10$ e $k = 15$.

No sistema TuriCreate, as recomendações são geradas recebendo um código identificador de usuário como entrada, de forma que, para cada pessoa usuária que consta na base de treino, o sistema tenta gerar sugestões de itens relevantes para ela. Para fins de avaliação do sistema, utilizando a proposta da métrica HR@K, 80% dos itens constados no histórico de interações dos usuários foi utilizado para treino, e a avaliação se deu de forma que, para cada sugestão gerada para os usuários, se pelo menos um item sugerido constava nos 20% do histórico de interações não utilizado para treino, um acerto era contabilizado. As tabelas abaixo contém os desempenhos de todos os sistemas, os valores em negrito apontam os maiores valores alcançados. O desempenho do TuriCreate com ambas as bases de dados é encontrado na tabela abaixo.

Tabela 4: Resultados obtidos com o Turicreate (Similaridade de Jaccard).

Base de Dados	HR@15	HR@10	HR@5
MovieLens	15,27%	15,27%	15,12%
Metacritic Game Review	13.79%	13.78%	13.77%

No sistema D-Mean, que gera sugestões de itens recebendo um ou mais itens de entrada, foram geradas sequências de itens para cada usuário, 80% destas sequências foram utilizadas para os cálculos de similaridade, enquanto os 20% restantes foram utilizados para teste, de forma que cada sequência foi dividida na metade, a primeira metade foi enviada para o sistema como "produtos de entrada", e a segunda metade foi utilizada como sequência alvo para o cálculo do desempenho utilizando HR@K. Os resultados obtidos são ilustrados na tabela abaixo.

Tabela 5: Resultados obtidos com DMean

Base de Dados	HR@15	HR@10	HR@5
MovieLens	15,83%	10,81%	8,25%
Metacritic Game Review	7,42%	6,8%	4,3%

No que diz respeito aos modelos referidos no trabalho como LSTM e LSTM-TSTM, por também serem modelos que geram recomendações considerando os itens indepen-

dente do usuário, ambos receberam os dados arquitetados da mesma maneira do sistema D-Mean: os itens interagidos foram agrupados por usuários e 80% desses históricos de interação foram utilizados para treino, e 20% foram utilizados para teste. De cada histórico, a primeira metade de itens foi interpretada como entrada, e a segunda metade foi interpretada como alvo para o cálculo do desempenho utilizando a taxa de acerto HR@K. Os resultados do LSTM e do LSTM-TSTM são ilustrados abaixo:

Tabela 6: Resultados obtidos com LSTM

Base de Dados	HR@15	HR@10	HR@5
MovieLens	54,84%	38,53%	13,68%
Metacritic Game Review	42,82%	42,82%	27,30%

Tabela 7: Resultados obtidos com LSTM-TSTM

Base de Dados	HR@15	HR@10	HR@5
MovieLens	90,3%	90,3%	60,98%
Metacritic Game Review	20,1%	20,1%	9,58%

Em todos os casos, os resultados indicam que, utilizando a base MovieLens, com históricos de filmes assistidos pelos usuários, os sistemas tiveram desempenho estatisticamente melhor do que utilizando a base de jogos. Este comportamento dos sistemas, entretanto, pode se dar pelo fato de que embora a base Metacritic Game Review possua uma quantidade alta de históricos de compras, muitos usuários interagiram com poucos jogos, de forma que a grande maioria das sequências a serem interpretadas pelos sistemas acabavam sendo curtas, e foi observado que quando as sequências são longas (20 itens pra cima) ou curtas (5 itens pra baixo), o desempenho dos sistemas caía, especialmente os que faziam uso das redes LSTM.

Observando as sugestões das redes que tiveram melhor desempenho, foi possível notar que muitas das vezes em que as recomendações que eram classificadas como erradas pela métrica HR@K (por não possuir nenhum item alvo nas recomendações), havia pelo menos um item muito similar a um dos itens alvos, como no exemplo ilustrado pela figura 10. Esse comportamento aponta o quão severa a métrica de avaliação HR@K é, visto que para um sistema de recomendação, a sugestão não precisa ser necessariamente igual ao item alvo, afinal, a proposta de um sistema de recomendação é indicar itens relevantes para um usuário.

Figura 10: Exemplo de recomendações erradas feitas pelo LSTM-TSTM com a base Metacritic Game Review.

Recomendação	Filme de Entrada	Filme Esperado	Filme Recomendado
<ul style="list-style-type: none"> • Número 1 • Número 2 	<ul style="list-style-type: none"> • The Legend of Zelda: Twilight Princess • Uncharted 2: Among Thieves 	<ul style="list-style-type: none"> • Super Smash Bros. for Wii U • Uncharted: Drake's Fortune 	<ul style="list-style-type: none"> • Super Smash Bros. Brawl • Uncharted 3: Drake's Deception

Fonte: Elaborado pela autora

Ainda sim, mesmo com as limitações encontradas, analisando os resultados obtidos com os modelos LSTM e LSTM-TSTM, é possível perceber que o desempenho dos modelos foi satisfatório, considerando especialmente resultados de modelos encontrados no estado da arte. Os resultados obtidos por Sun and Qian (2020), que testaram seus modelos em 2 bases, uma delas também sendo uma amostra da fonte de dados do MovieLens e a outra sendo uma amostra da fonte de dados do Gowalla (dados de uma rede social de *check-in* em locais), e obtiveram resultados de HR@K para a amostra do MovieLens variando de 21% (HR@5) até 59% (HR@15), e para a amostra do Gowalla variando de 47% (HR@5) até 81% (HR@15).

O modelo modificado do LSTM-TSTM conseguiu uma taxa de acerto bastante positiva para a amostra da base do MovieLens, e o modelo simples LSTM conseguiu uma taxa boa para a amostra da base do Metacritic Game Review, considerando o modelo original proposto por Sun and Qian (2020) e também que o objetivo dos sistemas é gerar recomendações relevantes, ou seja, recomendações de itens com os quais o usuário gostaria de interagir, e a métrica avalia apenas quantas dessas recomendações condizem com o que o usuário interagiu na vida real. Partindo dessa perspectiva e considerando as taxas de recomendações relevantes, dentro de um top 10, acima de 40% para a base de jogos com o LSTM, e acima de 90% para a base de filmes com o LSTM-TSTM, é seguro afirmar que esses sistemas conseguiram alcançar um ótimo desempenho no contexto de recomendações de itens utilizando dados implícitos, a figura 11 ilustra um exemplo de recomendação de sucesso do LSTM-TSTM com os dados do MovieLens, na figura, os acertos são indicados pelos títulos em negrito.

Figura 11: Exemplo de recomendação de sucesso feita pelo LSTM-TSTM com a base MovieLens

Filmes de Entrada	Filmes Esperados	Filmes Recomendados
<ul style="list-style-type: none"> • Leaving Las vegas (1995) • Usual Suspects, The (1995) • Antonia's Line (Antonia) (1995) • Braveheart (1995) • Apollo 13 (1995) • Madness of King George, The (1994) • Natural Born Killers (1994) • Pulp Fiction (1994) • Shawshank Redemption, The (1994) • Star Trek: Generations (1994) 	<ul style="list-style-type: none"> • Adventures of Priscilla, Queen of the Desert, The (1994) • Forrest Gump (1994) • True Lies (1994) • Fugitive, The (1993) • Jurassic Park (1993) • Schindler's List (1993) • Dances with wolves (1990) • Batman (1989) • Silence of the Lambs, The (1991) • Bread and Chocolate (Pane e cioccolata) (1973) 	<ul style="list-style-type: none"> • Jurassic Park (1993) • Tommy Boy (1995) • Searching for Bobby Fischer (1993) • Home Alone (1990) • Aladdin (1992) • Dances with wolves (1990) • Batman (1989) • Beauty and the Beast (1991)

Fonte: Elaborado pela autora

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

O presente trabalho apresentou quatro sistemas de recomendação de itens afim de demonstrar que é possível gerar recomendações relevantes baseadas em dados implícitos, possibilitando a democratização do uso de sistemas de recomendação para todos os pequenos, médios e grandes negócios que optam por não utilizarem dados possivelmente sensíveis, como dados demográficos e/ou preferenciais de seus clientes.

Considerando os estudos realizados e os resultados levantados, é possível afirmar que, para a base de dados de filmes (MovieLens), o sistema de recomendação utilizando uma arquitetura LSTM-TSTM demonstrou-se expressivamente satisfatório; e para a base de dados de jogos (Metacritic Game Review), o sistema de recomendação utilizando uma arquitetura LSTM teve resultados também positivos, embora ainda limitados.

Analisando a complexidade do trabalho, a parte mais desafiadora do projeto se deu no momento de decidir quais arquiteturas implementar e quais adaptações fazer com os dados devido a restrição de hardware, uma vez que as bases eram bastante volumosas e esparsas, e os hardwares utilizados não eram de ótimo desempenho. Outro pico de complexidade se deu no momento de analisar os resultados e perceber a severidade da métrica de avaliação, afinal, para obter tal discernimento, foi necessária uma abstração do pensamento técnico mais crítico e uma adoção de um pensamento mais humano em relação à tomada de decisões. Mesmo com limitações, ambos os resultados obtidos indicam que há um caminho longo e promissor a ser construído no que diz respeito à implementação de sistemas de recomendação com dados implícitos. Enxergar a possibilidade de criar recomendações relevantes rapidamente com dados não sensíveis fez da construção do trabalho uma jornada bastante empolgante.

7.2 Trabalhos Futuros

No que diz respeito a evolução do trabalho, os próximos passos a serem tomados envolvem o estudo de métricas de avaliação menos severas, considerando a semelhança das recomendações com o histórico de interações do usuário, e não necessariamente a presença dos itens previamente interagidos pelo usuário nas recomendações. Nesse aspecto, também aplicar um teste de hipótese afim de considerar se manter um top 15, que possui as maiores taxas, é realmente mais vantajoso do que um top 10, que possui taxas muito similares. Além disso, seria de extrema relevância o retreino dos modelos bem sucedidos no trabalho com um volume maior de dados, especialmente com dados de compras, visando compreender o desempenho dos sistemas para aplicações de varejo. Por último, a

continuação da busca por modelos de sucesso para recomendações com dados implícitos, visto que o campo do aprendizado de máquina segue em constante evolução.

REFERÊNCIAS

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.
- Alsini, A., Huynh, D. Q., and Datta, A. (2020). Hit ratio: An evaluation metric for hashtag recommendation. *CoRR*, abs/2010.01258.
- Brovman, Y. (2019). Complementary item recommendations at ebay scale. <https://tech.ebayinc.com/engineering/complementary-item-recommendations-at-ebay-scale/>. Accessed: 2020-11-02.
- Chen, D. (2020). Recommender System - Matrix Factorization. <https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b>. Accessed: 2021-04-26.
- Chiavenato, I. (2005). *Administração nos novos tempos*. Elsevier Editora.
- Facure, M. (2017). Funções de Ativação: Entendendo a importância da ativação correta nas redes neurais. <https://matheusfacure.github.io/2017/07/12/activ-func/#fun-act>. Accessed: 2021-04-28.
- Ferro, M. R. d. C. (2010). Modelo de sistema de recomendação de materiais didáticos para ambientes virtuais de aprendizagem. Mestrado, Universidade Federal de Alagoas, Alagoas.
- Fonseca, J. J. S. d. (2002). *Apostila de metodologia da pesquisa científica*. João José Saraiva da Fonseca.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.
- Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Kang, W.-C. and McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

- Kingma, D. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kunchaparthi, S. (2021). Impact of covid 19 on e-commerce. *Journal of Interdisciplinary Cycle Research*, 12:1161.
- Lima, J. A. (2012). Liderança e tomada de decisão na organização. Graduação, Universidade do Oeste de Santa Catarina, Santa Catarina.
- Magalhães, C. M., Araújo, C. G., de Oliveira, E. T., and de Souza, H. L. (2013). O boom da internet e o impacto na relação entre empresas e consumidores. In *Bauru: XVIII Congresso de Ciências da Comunicação na Região Sudeste*.
- Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820.
- Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.
- Rocca, J. (2019). Understanding variational auto-encoders (vae). <https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>. Accessed: 2020-11-02.
- Sant’ana, V. A. d. (2016). E-commerce: Fatores que influenciam a satisfação em compras on-line na percepção dos graduandos de administração da ufsc. Graduação, Universidade Federal de Santa Catarina, Santa Catarina.
- Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.
- Souza, D. M. d. (2013). Modelos ocultos de markov: uma abordagem em controle de processos. Graduação, Universidade Federal de Juiz de Fora, Juiz de Fora.
- Sun, K. and Qian, T. (2020). Seq2seq translation model for sequential recommendation. <https://arxiv.org/abs/1912.07274>. Accessed: 2021-01-17.
- Yun-tao, Z., Ling, G., and Yong-cheng, W. (2005). An improved tf-idf approach for text classification. *Journal of Zhejiang University-Science A*, 6(1):49–55.
- Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38.