Sistema de Recomendação de Filmes por Sinopse com o Auxílio de Humano no Loop

Bruno de Santana Braga Contreras
11208072

Caio Rodrigues Gomes
11208012

Glaucia Pamponet Sobrinho
11271000

***Abstract*—This work proposes a recommender that uses vectors, acquired through the vectorization of content synopses to represent themselves, and that uses vectorial distance to obtain similarities, and collecting feedback from users in an Human-In-The-Loop scheme, pushing and pulling the vectors as necessary to improve the model's quality. It is discussed the methods and thinking behind this implementation, as well as the challenges faced by this model, such as the measures for performance evaluation and its current found flaws.**

I. Introdução

O uso da mineração de dados e de ferramentas de *Machine Learning* são atualmente um importante alicerce em diversos serviços consumidos de forma virtual ou tecnológica. Citando algumas das técnicas encontradas no mercado atualmente, é possível enfatizar algoritmos que definem perfilamento para a discriminação de dados manipulados, ou regras de associação que definem padrões de relacionamento e predições que apresentam uma aproximação adequada para um dado pertencente ao conjunto submetido através de imagens, sons e textos.

A mineração de textos está presente em mais locais do que podemos imaginar. Encontramos esse tipo de ferramenta nas entradas de dados textuais em tratamentos de pesquisas feitas na internet, interpretação em assistentes virtuais, *chat bots* e recomendações de produtos, o que acaba se tornando um grande investimento e acerto para os comércios, principalmente os virtuais, uma vez que a aplicação de maleabilidade ao uso das ferramentas digitais de acordo com a necessidade do usuário garante maior satisfação e melhor experiência de uso das plataformas. Entradas de dados textuais, por possuir valor semântico, acabam solicitando mais recursos voltados a virtualização de seu significado e menos uso de manipulação matemática no dado em si. Considerando justamente a aproximação de precisão do significado que algoritmos manipuladores desse tipo de dado devem almejar, a digitalização de uma única palavra para ser processada em um programa responsável por categorizá-la ou descobrir sua associação a um contexto ou a outra palavra deve possuir mais detalhes do que os necessários para se processar um dado quantitativo. Esse é o caso das tecnologias responsáveis por descobrir e rotular medidas de relações entre palavras e sentenças, como no caso dos *Word Embeddings*.

Word Embeddings se mostra um instrumento de grande efeito no processamento de linguagem natural. O uso de *deep learning* é usado na técnica para a aprendizagem de associação de palavras dentro de sentenças as quais elas estejam incluídas. O produto das tarefas de associação de palavras e sentenças pode resultar em ferramentas extremamente úteis para analise de discursos com o uso da vetorização realizada na técnica. Os métodos mais comuns envolvem o traçamento de distâncias entre os objetos por meio de modelos de *machine learning*. Essas medidas podem ser usadas para observar e apontar aproximações entre esses objetos de forma que consigam identificar semelhanças. Entretanto, como se trata de cálculo de semelhança de valor semântico e interpretativo, a análise dessas aproximações podem reproduzir respostas equivocadas, que dependem de treinamento específico.

O projeto proposto por este grupo aborda um uso comercial bastante comum das técnicas de *Machine Learning* envolvendo a captura, processamento de identificação de padrões no uso de serviços e na compra de produtos para que ofereçam sugestões satisfatórias que mantenham constante o uso das plataformas digitais, os algoritmos de recomendação. Se tratando e uso convencional, a execução desse tipo de sistema é efetuado através de Filtragem Colaborativa, frequentemente baseada no perfilamento de usuário, onde registros de comportamento ao dado definem o conjunto de sujestão, ou Filtragem de Conteúdo, onde a similaridade do dado é a ferramenta responsável pela definição dos conjuntos de sugestão. O modelo anterior será o abordado neste projeto usando um banco de filmes e sinopses com o uso de *Word Embeddings* para calculo de semelhança. Essa é uma abordagem que depende de um resultado adequado para o usuário da plataforma. Assim, entende-se que é necessário o uso de *feedback* visando regular o resultado para garantir maior precisão.

II. Definição do Problema

De acordo com [1], seja U um conjunto de usuários que avaliam um filme favorito em uma plataforma, F o conjunto de filmes disponíveis na mesma plataforma. Existe uma função de recomendação r que determina o quão útil é para um usuário $u \in U$ consumir e favoritar um subconjunto selecionado de filmes de F com base em seu favorito. A função r deve atuar no plano U x F e deve produzir R, que consiste na escolha entre os filmes do conjunto F baseado no usuário u.

No problema discutido neste projeto, o conjunto F é munido de características para cada item f presente nele. A função r

deve compreender a busca dos maiores resultados comparativos entre características de todo $f \in F$, com o conteúdo de F favoritado por u, caracterizando uma filtragem por conteúdo.

A. Classificação das Sinopses

Em *Word Embeddings*, a análise das sentenças condiz com o sumário da etapa de definição das similaridades semânticas das palavras e as representações dos vetores de grandes dimensões, onde palavras que estão em mesmo contexto podem ser consideradas similares em significados e aparecerem próximas em um espaço dimensional. Na figura 1, podemos ver de forma simplificada e com menos dimensões nos vetores uma representação de como as palavras podem se relacionar semanticamente e apresentarem maior proximidade dentro do espaço dimensional das *embeds*.

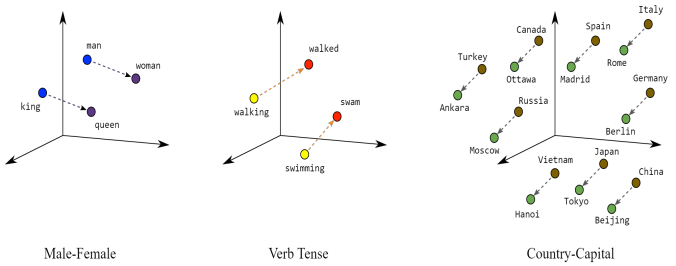

Fig. 1. Exemplo de espaços vetoriais onde as palavras foram distribuídas de acordo com os contextos semânticos.

Levando ao contexto do projeto produzido, podemos considerar as aproximações semânticas para definir o nível de similaridade dos textos definidos para os filmes armazenados no *dataset* de uso do trabalho. Entretanto, as *Word Embeddings* como as usadas no projeto, classificam apenas níveis de similaridade por palavras e sentenças, o que representa um nível abaixo do necessário para estipular a semelhança entre um texto de sinopse e outro.

B. Satisfação de Utilidade

Como J. J. Kamahara, T. Asakawa, S. Shimojo and H. Miyahara [2], duas possíveis abordagens eram possíveis: Um sistema de recomendação individual, ou um sistema generalizado o qual pudesse intercalar com interesses de outras pessoas. Devido às conclusões do mesmo artigo, que inv, escolhemos desenvolver um estilo de sistema de recomendação o qual é moldado aos interesses e interações de cada usuário.

Os resultados da transformação da função de recomendação acerca de cada usuário são comprometidas em retornar os elementos mais corretos no contexto matemático. Contudo, considerando o fator contextual e semântico, apenas os cálculos de maior semelhança de sinopses não venham a ser suficientes para considerar a satisfação de utilidade da função.

Como pontuado por David Chang [3], normalmente se tem mais dados do que os utilizados neste ensaio para se fazer uma recomendação mais embasada. São utilizados dados como o país de origem do usuário, dispositivo, horário de acesso e histórico de visualizações anterior, o que é sumarizado como ”Dados de Contexto”. Dito isso, parte do desafio é conseguir integrar esse tipo de informação a partir do Humano no Loop, mesmo que de uma forma limitada.

O uso de Humano no Loop, como dito em [4] constitui 5 tarefas com auxilio de ação humana para maior performance de *pipelining* dos dados (extração, integração, limpeza, rotulação interativa ou inferência). É possível facilmente destinar a atuação humana para rotulação dos dados ou da inferência no auxilio do treinamento do algoritmo, uma vez que a utilidade dele se baseia na experiência do usuário com o dado. Porém não é possivel esquecer que o trabalho humano no processo do loop tem seus empecilhos. Considerando o projeto em questão, é válido destacar a subjetividade na escolha dos resultados do sistema, uma vez que a falta de conhecimento do conteúdo dos dados ou as escolhas de gosto podem prejudicar o julgamento de resposta. Além disso, o fator performance traça um paralelo notável com os treinamentos supervisionados utilizando dados de teste, o que torna a latência ponto de atenção para as atividades do humano no loop.

III. Arquitetura da Solução

O fluxo da arquitetura da solução está exibido na Figura 2. Esse fluxo descreve qual o ciclo de vida da aplicação, elencando os principais pontos para seu funcionamento. O fluxo está dividido em 3 áreas, a área do usuário, que se faz relação com as ações que o humano pode executar no sistema. A área do Sistema de Recomendação, chamada de *RecSys*, é responsável pela lógica de receber o filme do usuário, retornar o filme recomendado pelo sistema e gerenciar a avaliação do usuário. Por fim a área da base de dados, essa área é responsável por armazenar como os filmes serão recomendados, armazenando uma matriz de distâncias e um espaço vetorial que será discutido nas próximas seções.

Partindo da interação do usuário com o sistema, o usuário seleciona um filme de seu agrado, que esteja presente no conjunto de filmes da aplicação. Após selecionar o filme, o sistema busca em sua matriz de distâncias, qual o filme mais próximo do filme selecionado, que não seja o filme escolhido pelo usuário, retornando essa busca para o usuário. Por fim a última interação do humano, faz ele entrar no loop da aplicação, uma vez que o usuário avalia, se a recomendação do sistema foi uma boa recomendação ou não, fazendo o sistema lidar com essa situação, ajustando internamente sua lógica, para reforçar a recomendação caso a avaliação seja positiva, ou mitigar, caso a avaliação seja negativa.

A. Matriz de Distâncias e Espaço Vetorial

Para que o sistema consiga entregar recomendações suficientemente satisfatórias, foi utilizado a técnica de armazenar cada filme em um vetor, assim gerando um espaço vetorial com todos os filmes do conjunto de dados. Cada filme terá uma coordenada vinculada a ele, e cada uma dessas coordenadas do filme descreve o mesmo em algum aspecto, não definido, utilizando o mesmo conceito do *word embedding*. Por sua vez um filme terá uma coordenada de n dimensões, assim extraindo

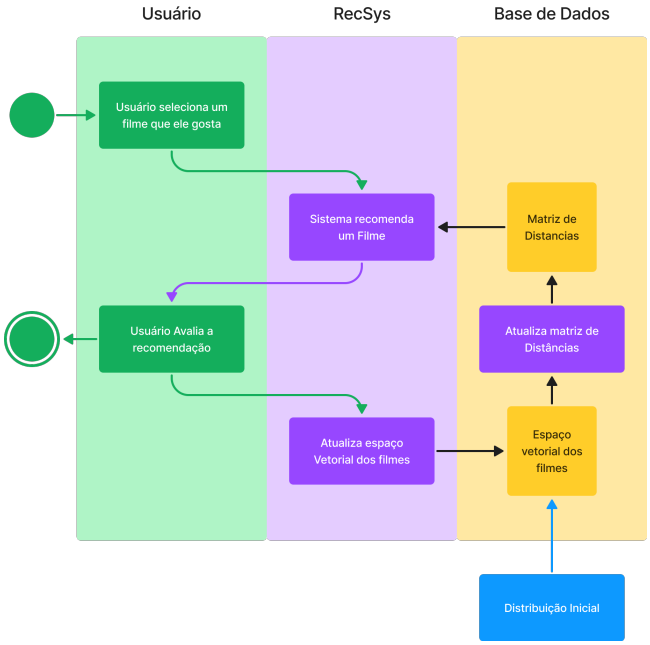

Fig. 2. Fluxo do sistema de recomendação com Humano no Loop

mais *features*, na tentativa de aumentar a assertividade da recomendação.

Para a extração desses vetores, foi-se processado o texto através do modelo *skip-gram Word2Vec*, que de acordo com Mikolov Et al. [5], é um método eficiente que gera vetores de alta qualidade através de grandes quantidades de texto não estruturado.

A Figura 3 representa o espaço vetorial dos filmes, de forma simplificada, utilizando apenas duas dimensões.

A matriz de distâncias é responsável por descrever a distância entre todos os filmes entre si. Então é criado uma matriz n x n, sendo n o número de filmes existentes no conjunto de dados. A princípio para realizar o cálculo da distância foi utilizado a distância euclidiana, exibido na Equação 1. Entretanto a utilização de outras distâncias é totalmente passível de utilização. A Figura 4 exibe uma matriz bidimensional das distâncias do espaço vetorial exibido na Figura 3.

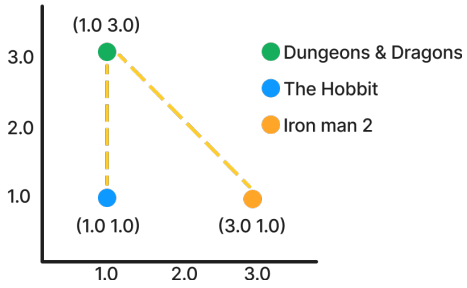

Fig. 3. Exemplo do espaço vetorial com duas dimensões, com 3 filmes exibidos

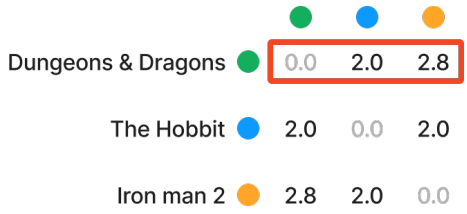

Fig. 4. Exemplo de matriz de distâncias entre todos os filmes, utilizando a distância euclidiana

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \quad (1)$$

Com a matriz de distâncias construídas é possível verificar qual o filme mais próximo do filme selecionado. No exemplo da Figura 4, o filme selecionado é *Dungeons & Dragons*, ao observar a linha de distâncias deste filme observa-se o vetor $(\,0.0\ 2.0\ 2.8\,)$, utilizando ele é possível selecionar a posição com o menor valor diferente de 0. Nesse caso o filme mais próximo é *The Hobbit*, consequentemente o sistema retorna para o usuário o filme recomendado com a menor distância, ou seja, com a sinopse mais similar.

B. Distribuição Inicial

Inicialmente o conjunto de dados não possui nenhuma representação vetorial dos filmes, dessa forma é preciso atribuir uma posição inicial para cada filme. Para esse feito é utilizado o conceito de *word embedding*, como mencionado na Introdução. É utilizando as sinopses dos filmes como fonte de dados textuais, gerando um corpus textual do conjunto de dados. Esse corpus então é submetido a um *word embedding*, a fim de criar um espaço vetorial de todas as palavras existentes no corpus do problema. Após esse passo, para cada filme, existe um conjunto de palavras que o representa, então é calculado a média desses vetores que representam palavras, assim resultando em uma única posição vetorial por filme. Por fim, o produto final desta operação é um conjunto chave-valor com cada título do filme vinculado à uma posição no espaço vetorial. A Figura 14 representa como esse fluxo de distribuição inicial deve se portar. Note que as únicas informações extraídas do conjunto de dados do filme são o título e a sinopse.

C. Humano no Loop

O momento do humano no loop ocorre logo após da recomendação do filme mais relevante, coletando o *feedback* do usuário. Essa coleta se trata de uma coleta binária, ou seja, após a recomendação do filme, pede-se que o usuário avalie se a recomendação atendeu as expectativas dele, ou não. Entrando assim em 2 cenários, onde o sistema, sim, conseguiu atender as expectativas do usuário, fazendo com que essa recomendação seja reforçada dentro do sistema. Ou o cenário de não, o sistema não fez uma boa recomendação, fazendo com que o sistema haja para que essa recomendação seja menos recorrente.

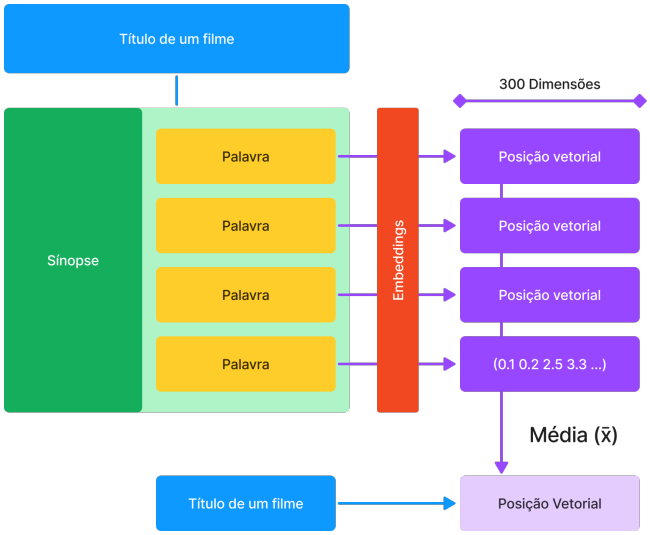

Fig. 5. Fluxo da geração da primeira instância de posições vetoriais.

Cenário SIM: Para o cenário que sim, foi uma boa recomendação. A posição do filme recomendado irá se aproximar do filme selecionado. Aumentando a chance dos filmes serem recomendados juntos. Essa aproximação ocorre de forma parametrizada, utilizando a Equação 2.

Cenário NÃO: Para o cenário que não, o filme recomendado não é uma boa recomendação, o filme recomendado se distanciará do filme selecionado pelo usuário. Assim diminuindo a chance do filme ser recomendado ao selecionar o filme selecionado pelo usuário. O distanciamento do filme ocorre utilizando a Equação 3.

$$N = s \cdot (O - A) + A \tag{2}$$

$$N = (1 + s) \cdot (A - O) + O \tag{3}$$

Para ambas as equações, seja N a nova posição do filme recomendado, A a posição atual do filme recomendado, O o filme que o usuário selecionou originalmente e s o parâmetro de sensibilidade da aproximação ou distanciamento, ou seja, o quanto o filme irá ser deslocado em comparação com a distância entre o filme selecionado e o recomendado. O parâmetro s pode possuir valores entre 0 e 1, no caso da aproximação, caso o valor de s seja 1, a nova posição do filme recomendado, será a mesma do filme selecionado pelo usuário. Enquanto que no caso do distanciamento, caso o valor de s seja 1, a nova posição do filme recomendado será duas vezes a distância entre O e A. Para ambos os casos, se $s = 0$, a posição do filme recomendado não se altera. Valores intermediários, representam um percentual de deslocamento.

D. Base de Dados

A fonte de dados escolhida para a alimentação do sistema foi retirada do estudo de atribuição automatizada de tags a filmes de [6][1]. Trata-se de um *dataset* que armazena registros de filmes, séries e programas de televisão, apresentando numero identificador, título, texto relacionado a sua sinopse ou apresentação do conteúdo e outros campos que se mostraram irrelevantes ao estudo apresentado aqui.

IV. Configuração de Testes

O objetivo dentro da observação da estrutura no sistema de recomendação consiste em perceber o impacto da inserção de ação humana no auxilio da escolha de conteúdo sugerido ao usuário na execução do algoritmo. Dessa forma, o conjunto de testes está apoiado em ser executado de forma que apresente as diferenças e comparações entre a estrutura não interferida por usuário de uma estrutura que veio a sofrer modificações após o uso por um grupo com uma determinada quantidade de usos ou iterações. Os testes estão divididos entre ações de experimentação e ações de observação ou de coleta de métricas possíveis, em que avaliaremos o estado da estrutura vetorial do sistema.

A. Ação de Ambientação

Considerando o uso de um sistema de recomendações que atende a um determinado grupo, a ação de ambientação consiste em estabelecer o uso da ferramenta por um grupo de 3 a 5 pessoas. O grupo deve possuir um espaço vetorial inicial para cada usuário, a fim de impedir conflitos na posição vetorial, uma vez que para cada escolha de preferência, a posição dos vetores representantes das sinopses dos filmes do *dataset* tende a transitar.

Cada usuário deve completar 20 ciclos de uso do algoritmo, em que cada ciclo consiste em:

1) Pesquisa de um título existente na base;
2) Avaliação das sugestões trazidas pelo algoritmo.

B. Ações de Observação

Os testes de observação foram definidos visando acompanhar o comportamento da malha vetorial em situações específicas após as ações de iteração envolvendo escolha humana. O trabalho na movimentação estrutural do espaço vetorial consiste em:

a) Precisão: consiste na precisão através do tempo das recomendações sobre um determinado filme. A definição de ”precisão” aqui considerada é a taxa de respostas aceitáveis para uma certa iteração.

b) Distancia: O segundo teste visa comparar e tentar descobrir se há diferença na velocidade de convergência das respostas caso se utilize uma função para metrificação da distância vetorial diferente.

c) Vizinhança: O terceiro teste visa observar como que as recomendações afetam os vetores mais próximos e suas recomendações.

[1]https://www.kaggle.com/datasets/cryptexcode/mpst-movie-plot-synopses-with-tags

V. Resultados Obtidos

As distâncias vetoriais são reiniciadas para seu valor inicial no início de cada teste, tal como veem do método de vetorização.

A. Teste de Precisão

Para este teste, foi escolhido um filme aleatório, avaliadas as recomendações obtidas visando filtrar ao gosto do usuário até que todas as recomendações fossem um *hit*, que consiste em 100% de aprovação, de acordo com a interpretabilidade do avaliador.

No nosso caso de teste, utilizamos o filme ”Homem de Ferro”, um filme de ação e ficção científica de super-herói. Assim, o algoritmo foi recomendando filmes na vizinhança até que se encerasse o teste. Na primeira recomendação, o algoritmo já sabia indicar as duas sequências de Homem de Ferro, o que são filmes os quais pessoas que assistiram o primeiro muito provavelmente vão querer ver, devido à similaridade de seus enredos. Nas iterações seguintes, filmes como ”Os Vingadores”, ”O Dia da Independência” e ”O Cavaleiro das Trevas Ressurge” começam a aparecer conforme as recomendações não desejadas são negativamente avaliadas e afastadas do vetor de ”Homem de Ferro”.

Sendo 6 recomendações por iteração, as recomendações começam com uma taxa de 33% de porcentagem de hit, aumentando para 66% pela sétima iteração, chegando em 83% pela nona iteração e 100% pela décima iteração.

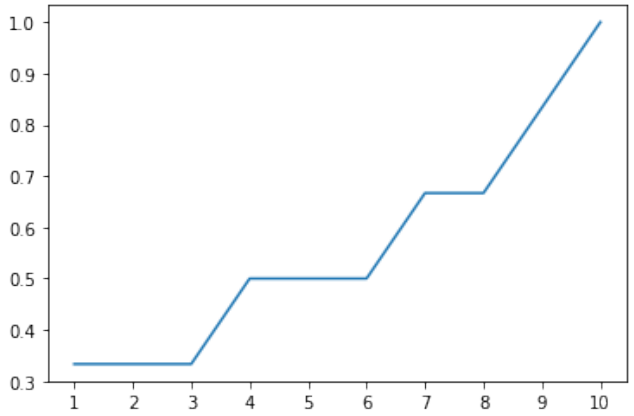

Fig. 6. Exibição da precisão do caso específico conforme iterações.

A métrica para as relações das *embeds* que integra a precisão sofreu alterações após avaliação submetida a interpretabilidade humana. A figura 6 exibe a formação das recomendações do filme ”Moana” após as iterações de humano no loop.

Foram encontrados resultados de submissões pós-humano-no-loop apresentando precisões relativas, como o caso da figura 7. O tratamento da interatividade humana gerou 83% das indicações adequadas, o que resultou em um filme restante considerado como sugestão não adequada de acordo com a interpretação humana, mas sim para o algoritmo, considerando sua arquitetura.

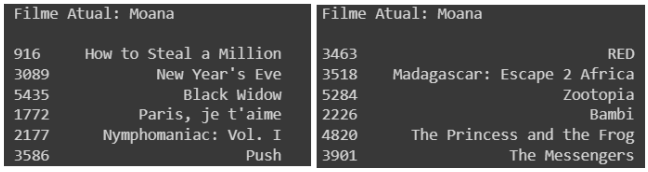

Fig. 7. Exibição de recomendação a filme antes (esq.) e após interatividade humana na submissão (dir.)

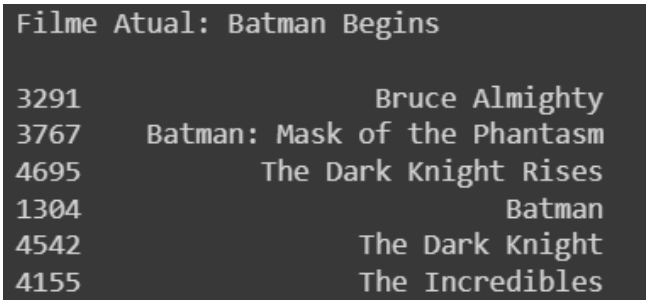

Fig. 8. Sugestões ao filme Batman após interatividade

Também checamos a porcentagem de *hits* da recomendação inicial através da distância por cosseno.

Neste teste, vimos a recomendação de 20 filmes sem alterar valores vetoriais, assim contando o número de *hits* inicial, com o objetivo de obter um parâmetro para comparação de melhoria.

Assim, obtivemos como resultado 64 *hits* iniciais, dentro 120 recomendações iniciais, o que nos dá uma precisão de 53% inicial.

B. Teste de Função de Aproximação

Até o primeiro teste, o uso de função definidora das diferenças de distâncias vetoriais adotadas nas recomendações era a função de distância euclidiana. A partir desse patamar, a nível de comparação, foi definida como função de valor de distância a função por cosseno.

Para treinar interativamente o algoritmo com base na distância por cosseno, foram usados os mesmos 21 filmes treinados no uso do algoritmo com base em distância euclidiana. A figura 8 exibe as distribuições de iterações necessárias em ambas funções de aproximação até que se considerasse o resultado desejado como 100% de precisão para cada filme.

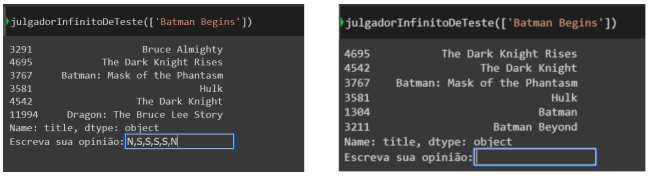

Fig. 9. Exibição de recomendação antes(esq.) e depois(dir.) de interatividade humana, porém agora utilizando distância por cosseno como método.

C. Teste de Vizinhança

Para este teste, foi utilizado um filme retirado da lista de treinamento executado nos testes anteriores. Para analisar o

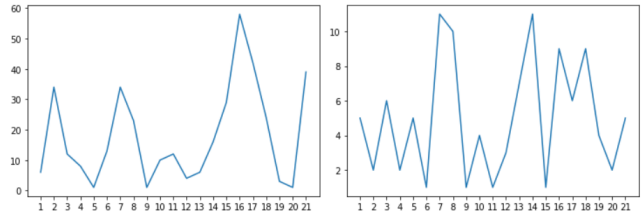

Fig. 10. Número de recomendações necessárias para que todas sejam hits, versão distância euclidiana (esq.) e versão distância por cosseno (dir.).

Filme Atual: Kick-Ass	
2548	Kick-Ass 2
1772	Paris, je t'aime
3500	2001: A Space Odyssey
4147	Watchmen
5543	Grindhouse
4041	Phantoms

Filme Atual: Kick-Ass	
2548	Kick-Ass 2
2994	How I Won the War
2010	The Hollywood Knights
4477	The Ladykillers
3102	MASH
3463	RED

Fig. 11. Sugestões do filme Kick-Ass antes (esq.) e após treinamento (dir.).

comportamento dos vetores proximos aos resultados emitidos para um filme específico, foi consultado o filme pertencente ao treinamento anteriormente ao *loop*. Após a consulta e o treino, foi refeita a mesma chamada de recomendações do filme treinado e de seus respectivos relacionados, de acordo com o julgamento humano.

Enquanto pré treino, foram coletadas as precisões declaradas pela classificação do humano no loop para cada consulta, assim como para pós treino. Nas exibições das figuras (n das figuras aqui), é possível observar o impacto da precisão do retorno do algoritmo e a presença dos itens relacionados à consulta e treinamento dentro da vizinhança do dado de treino.

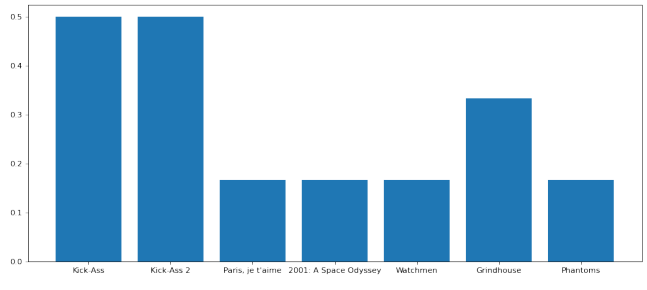

Fig. 12. Precisões dos filmes relacionados a Kick-Ass antes do humano no loop.

VI. Análise dos Resultados

Considerando a estrutura do projeto, é possível notar que o algoritmo permite uma performance excelente para filmes que possuem sequências. Isso se deve pela similaridade entre os enredos nas produções inerentes a sagas e franquias, o que facilita ao código reconhecer nomes de personagens, lugares, e situações dentro do roteiro da obra. Pensando na atuação do humano no loop na fase de treinamento interativo, o aparecimento de franquias nas sugestões tende a facilitar o

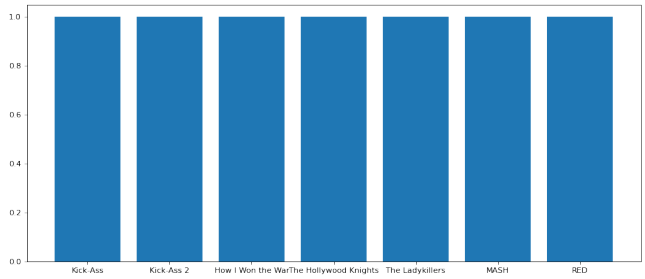

Fig. 13. Precisões dos filmes relacionados a Kick-Ass após humano no loop.

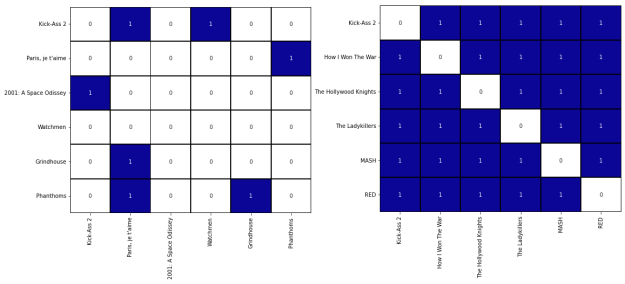

Fig. 14. Heatmap apresentando a presença de relacionados escolhidos no treinamento antes (esq.) e depois (dir.) do humano no loop.

julgamento das sugestões. Isso pode ser perceptível ao examinar as iterações necessárias de ação humana ao validar todas as 6 sugestões como adequadas para títulos como ”Frozen”, ”Moana” e ”Grease” (7, 16 e 17 nos gráficos da figura 8).

É importante ressaltar que alguns resultados de iterações, precisões iniciais e cálculos de distância sofreram interferência do conflito de idiomas e de conteúdo do *dataset* escolhido. Houveram ocorrências de campos de sinopse que pertenciam a resenhas e analise do filme referente, o que interferiu em quantidade de rejeições para sugestões na interação humana.

Outro caso notável a se observar referente à classificação dos vetores de média que representam os filmes no *dataset* usado são as recomendações de adequações relativas, como no caso do exibido na figura 7. Aqui é possível notar a divergência entre a definição de similaridade da classificação vetorial do algoritmo com a da interpretabilidade humana. Como mencionado, a média vetorial para cada item estabelece similaridades baseadas em características do roteiro, como a característica encontrada entre ”Bruce Almighty” e ”Batman Begins”, que não se trata especificamente de enredo da história.

Devido à diferença no total de iterações necessárias para a declaração de aceitação aos 100% de precisão para cada filme, as comparações entre uso da distancia euclidiana e distancia por cosseno são feitas considerando a proporcionalidade na submissão de cada filme a uma avaliação. Apenas considerando a informação anterior, é plausível atribuir à função de aproximação por cosseno como uma opção mais vantajosa em comparação a distância euclidiana. Em termos gerais, quando visto a performance da figura 8, fica comprovado que a distância euclidiana teve a maior amplitude, enquanto que

a distância por cosseno, que contou com mais variações de resultado do que o teste com o uso da distancia euclidiana, conseguia chegar em seis acertos com um menor número de interações.

Assim como no comportamento de resultados do algoritmo usando a distância euclidiana, foi percebida muita influência dos resultados em franquias no uso da distância por cosseno. O comportamento da classificação ainda é sensível a nível de nomes característicos de roteiro, uma vez que o uso de calculo da distância não implica na transformação vetorial para similaridade. Dessa forma, a inferência produzida por essas observações é de que a medida de calculo da distância vetorial aplica pouco impacto na sensibilidade de interpretação das *word embeddings* calculadas em média para a representatividade dos filmes.

A vizinhança relacionada exibida nos últimos testes é interessante de se observar. Fica perceptível que há uma criação de nichos ao redor dos dados ao obterem feedback não artificial. A relação entre os sugeridos como seus próprios recomendados é compreensível, pois devem compartilhar da mesma área vetorial contextualmente falando, se pertencem ao mesmo grupo de enredo da história.

VII. Conclusão

Após estudo da analise dos testes realizados, é capacitável de se concluir a discrepância da interpretabilidade entre o conceito de sugestivo para um algoritmo treinado e o definido pela interação humana, que agrega maior valor, principalmente considerando o uso no cotidiano e no mercado. No caso do uso de semelhança semântica de conteúdo, como o projeto de recomendação baseado em filtragem de conteúdo apresentado aqui, fica claro a existência de limitações referentes à relação de interpretabilidade que a máquina e o ser humano adotam para o conteúdo. As escolhas de calculo das características inerentes ao processo de escolha dos itens de semelhança são importantes na tomada de decisão do projeto, mas pouco representam, caso a definição semântica e contextual do conteúdo não esteja bem definida antes da aplicação do cálculo.

References

[1] S. Ricci, Rokach, *Recommender Systems Handbook*. Springer, 2nd ed., 2015.

[2] J. Kamahara, T. Asakawa, S. Shimojo, and H. Miyahara, "A community-based recommendation system to reveal unexpected interests," in *11th International Multimedia Modelling Conference*, pp. 433–438, 2005.

[3] D. Chong, "Deep dive into netflix's recommender system." https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48. Accessed: 2022-07-20.

[4] C. Chai and G. Li, "Human-in-the-loop techniques in machine learning.," *IEEE Data Eng. Bull.*, vol. 43, no. 3, pp. 37–52, 2020.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[6] in *Proceedings of the Ano = 2018, mês = maio, data = 7-12, local = Miyazaki, Japão, editor = Nicoletta Calzolari (presidente da conferência).) e Khalid Choukri e Christopher Cieri e Thierry Declerck e Sara Goggi e Koiti Hasida e Hitoshi Isahara e Bente Maegaard e Joseph Mariani e Helen Mazo e Asuncion Moreno e Jan Odijk e Stelios Pipeperidis e Takenobu Tokunaga, editor = European Language Resources Association (ELRA), endereço = Paris, France, isbn = 979-10-95546-00-9, idioma = english*.