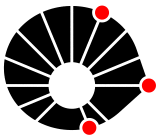
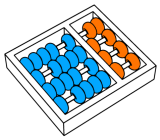




Sistema automatizado de questão e respostas em *e-commerce* baseado em similaridade de sentenças multilíngues.

Luiz Eduardo Araujo Zucchi, Julio Cesar dos Reis

Relatório Técnico - IC-PFG-21-14
Projeto Final de Graduação
2021 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Sistema automatizado de questão e respostas em *e-commerce* baseado em similaridade de sentenças multilíngues.

Luiz Eduardo Araujo Zuccho e Julio Cesar dos Reis*

Julho 2021

Resumo

Chatbots veem se tornando ferramentas cada vez mais essenciais para o atendimento ao cliente nas plataformas de *e-commerces* modernas. Responder perguntas de forma rápida e correta é em muitos casos a diferença entre realizar uma venda ou não. Conseguir um sistema de chatbot para responder de maneira adequada automaticamente em múltiplas línguas é um grande desafio de pesquisa. Este trabalho propõe utilizar um modelo de redes neurais chamado *DeepSim* para tratar e avaliar esse problema para as línguas Portuguesa e Espanhola. Em nossos experimentos, avaliamos o impacto de dados anotados na efetividade do modelo treinado para estimar o trabalho necessário para expandi-lo para outras línguas.

1 Introdução

O atendimento ao cliente é uma parte essencial do comércio eletrônico. Conseguir responder dúvidas de clientes de forma rápida e correta não só auxilia o cliente a fazer a compra, mas também ajuda ele(a) a confiar na loja, já que tipicamente existe certa ojeriza por parte de clientes com compras *online* devido ao grande número de golpes. Quando uma loja *online* em uma plataforma de comercio eletrônico responde a uma pergunta de um cliente de forma rápida e correta, o cliente se informa sobre o produto e passa a confiar mais na loja, o que por sua vez aumenta a chance dele comprar nela. Isso é relevante pois há uma vasta gama de opções (e concorrentes) no ambiente *online*.

A *GoBots* é uma das maiores empresas no segmento de chatbots para *e-commerce* na América Latina. Ela enfatiza soluções nesse ramo automatizando, através de

*Instituto de Computação, Universidade Estadual de Campinas, SP, Brasil

modelos de inteligência artificial, o processo de responder questões em plataformas de comercio eletrônico. A automatização ocorre de duas formas. A primeira consiste em classificar perguntas, usando modelos de classificação de intenção e extrair palavras chave das perguntas; e então com a intenção, e palavras chaves, processa uma lista buscando uma resposta que se adéque à questão do cliente. Outra solução consiste em usar uma busca semântica num banco de questões, específico por cada loja visando descobrir se aquela pergunta já foi feita antes e assim reutilizar a mesma resposta.

Atualmente, uma das dificuldades enfrentadas é que modelos existentes dependem da língua em que foram treinados originalmente para terem uma efetividade plausível. Um exemplo disso são modelos para classificar similaridade de perguntas visando reaproveitamento de respostas. Esta proposta vem se mostrando muito útil para lojas eletrônicas brasileiras, mas que não podem ser (re)usadas em *e-commerces* no restante da América Latina. Esses modelos foram originalmente treinados com sentenças na Língua Portuguesa. Além da questão da língua, outro problema é a dificuldade de se anotar dados para modelos no geral. Nesse contexto, seria útil ter uma estimativa do número de anotações minímas necessárias para conseguir usar o modelo em produção de forma confiável.

Neste trabalho investigamos um modelo de rede neural chamado *DeepSim* que classifica pares de perguntas como similares (ou não). A nossa proposta usa o modelo de codificação linguística *USE* [5] para gerar vetores que representam as perguntas,ou seja, um *encoder*, para então classifica-las. Além de desenvolver o modelo *DeepSim*, estudamos a capacidade desse modelo de classificar perguntas em outras línguas, além daquelas escritas na Língua Portuguesa, língua na qual o modelo foi originalmente treinado.

Em particular, investigamos o efeito da efetividade do modelo quando adicionamos novos dados anotados na língua Espanhola. Acreditamos que seja possível expandir a usabilidade do modelo para outras línguas com um pequeno esforço, graças ao fato do *USE* [5] ser um modelo multilinguístico, que já foi previamente treinado em várias línguas. Nossa proposta é demonstrar a possibilidade de reaproveitar o treino de um modelo efetuado numa certa língua em outra. Apresentaremos o esforço necessário para tal em termos de anotação de dados, que é uma tarefa que exige muitos recursos em diversas aplicações de aprendizado de máquina.

Em nossa metodologia, consideramos um conjunto de aproximadamente 3000 pares de perguntas na língua Espanhola, feitas por usuários reais da plataforma *Mercado Livre*. Esse conjunto foi recuperada do banco de dados da *GoBots*, de forma totalmente anonimizada para preservar a privacidade dos usuários. Os pares de perguntas são gerados de forma automática usando o método *KNN* [6], em que um par similar de perguntas consiste da pergunta mais próxima dela no *cluster* e qualquer outra. Para cada par similar recuperamos também uma não similar.

Esses pares foram manualmente revisados, e se necessário corrigidos como similares (ou não similares); por fim, revisados por terceiros não diretamente envolvidos

diretamente com esse trabalho. Com base nessa anotação de dados, esse conjunto foi tratado e separado entre treino, validação e testes. O conjunto de testes é usado para produzir os resultados finais demonstramos em nossos experimentos.

Além do conjunto de dados gerados na Língua Espanhola, usamos um conjunto de sentenças descritos na língua portuguesa com cerca de 20.000 pares de perguntas, anotado pela *GoBots* com dados obtidos em ambiente de produção; e um conjunto de dados em inglês obtido do *Quora Question Pairs* [18]. Esse é um conjunto de dados aberto na web em que perguntas de domínio aberto são classificadas como similares ou não similares.

Com esse conjunto de dados treinamos uma rede neural de classificação de duas formas: 1) uma delas foi usando um conjunto de dados misto, sendo parte dos dados em espanhol e o restante em outra língua; 2) a outra maneira foi treinando a rede "do zero", somente com os dados em uma única língua. Uma vez que visamos avaliar o impacto causado pela anotação dos dados, treinamos a rede neural usando parte do conjunto de dados de treino, e avaliamos o quanto a quantidade de dados nesse conjunto afeta a efetividade da rede no conjunto de testes (em espanhol). Os hiper parâmetros dessa rede são definidos através do algoritmo *Random Search* [17] que avalia diversos hiper parâmetros, dentro de limites pré definidos, buscando otimizar a acurácia do modelo. Esses hiper parâmetros são encontrados para a rede treinada somente com sentenças descritas na Língua Espanhola. Usaremos os mesmos hiper parâmetros em todos os testes, para que isso não afete nossos resultados.

Nossa investigação conduziu uma série de experimentos, como segue:

1. O primeiro é usado como uma *baseline* para os seguintes experimentos. Nele, avaliamos a rede neural treinada com sentenças na Língua Portuguesa no conjunto de testes em espanhol. Avaliamos a efetividade da rede sem nenhum tipo de treinamento adicional.

2. No segundo experimento treinamos uma rede usando somente o conjunto de dados com sentenças na língua Espanhola. Buscamos os melhores hiper parâmetros para a rede levando em conta o conjunto de dados completo. Então, usamos os valores encontrados para treinar a rede com sub conjuntos do conjunto de dados, verificando o impacto que o tamanho do conjunto apresenta na efetividade da rede.

3. O terceiro experimento considerou o mesmo procedimento que o primeiro. No entanto, este experimento usa dados na língua Portuguesa de duas formas: a primeira usa o mesmo numero de dados que em espanhol; e a segunda usa todos os dados em português disponíveis, cerca de 20.000 pares anotados. Usamos esses resultados como comparação para a rede treinada em espanhol. Objetivamos averiguar se misturar dados de uma língua com a outra aumenta a acurácia da rede.

4. O quarto experimento prosseguiu de forma similar ao terceiro. Contudo, usamos um conjunto de dados com sentenças descritas na língua Inglesa obtidos do *Quora Question Pairs* [18]. Avaliamos se o uso do inglês ao invés do português interfere na efetividade dos resultados da rede; e também se o uso de perguntas de outros domínios afeta os resultados.

O restante deste documento está estruturado da seguinte maneira: Seção 2 descreve os conceitos de base e trabalhos correlatos. Seção 3 descreve o modelo que construímos e o sistema em que o modelo é aplicado. Seção 4 detalha os experimentos e os resultados obtidos nas avaliações. Seção 5 discute os resultados obtidos e os impactos desta pesquisa. Seção 6 apresenta um fechamento do documento.

2 Revisão da Literatura

O uso de redes neurais para responder perguntas em linguagem natural vem sendo explorada na literatura de diferentes formas, e com diferentes especificações para o problema [14]. Esse problema no contexto de processamento de linguagem natural (PLN) é conhecido como *Question-Answering* [19]. Esse problema se divide em dois outros subproblemas mais específicos: 1) perguntas de domínio aberto, ou seja, perguntas que são sobre qualquer tipo de assunto, e que não requerem um conhecimento específico para serem respondidas [4] [7]. 2) perguntas de domínio específico ou fechado, que requerem que o sistema tenha conhecimentos específicos de um certo assunto.

A solução desenvolvida neste trabalho está alinhada com o segundo tipo de problema, pois nossa proposta (modelo de rede neural) leva em consideração certas especificidades das perguntas, para identificar se um certo par de perguntas são similares.

Nosso modelo proposto é integrado a um sistema de recuperação de informação com ranqueamento que visa responder questões de domínio fechado. Esse tipo de sistema visa utilizar conhecimento presente em respostas humanas (conhecidas pelo sistema) para responder questões futuras. Essa abordagem vem sendo amplamente usada em tarefas similares à que estudamos neste trabalho. Contudo, observamos que é necessário aprofundamento nos estudos quando o sistema precisa ser utilizado com múltiplas línguas

2.1 Conceitos Fundamentais

Em seguida descrevemos os conceitos relevantes para o entendimento de nossa proposta, sendo: *word embeddings*, *sentence embeddings*, *redes neurais siamesas*, *modelos multi linguísticos*; e por fim o conceito de *Busca Semântica*, que apesar de não ser diretamente o foco desse trabalho, é uma parte importante do sistema no qual nosso modelo é integrado.

Word embeddings são a base de várias aplicações de aprendizado de máquina para linguagem [1]. Eles consistem em um tipo de representação de termos, que permite que palavras que transmitem significados similares, tenham uma representação vetorial similar. Isso possibilita que, por exemplo, buscas em textos em linguagem natural sejam realizadas não só pela similaridade léxica das palavras, mas igualmente, pelo sentido (significado) em que são usadas. Além de permitir buscas mais aprimoradas, esse tipo de representação pode ser usada para o treinamento de redes neurais de forma mais efetiva, e cujo treinamento, faça mais sentido. *Word embeddings* podem ser gerados a partir de diversas técnicas diferentes, basta que essa técnica gere vetores que representam palavras num espaço vetorial pré definido.

Para obter essas representações, as técnicas existentes se baseiam em modelos de probabilidade [1], em que a representação distribuída de vetores de *features* são associadas a cada palavra. Cada uma dessas *features* são basicamente elementos do vetor, representam um aspecto dessa palavra. Por exemplo, verbos aparecem na língua Portuguesa em posições similares de uma frase, logo a representação vetorial de verbos obtém certas similaridades.

Podemos expandir o conceito de *word embeddings* para frases e documentos obtendo *sentence embedding* [11]. Isso se obtém através da média dos vetores de palavras que formam uma frase. Apesar de *word embeddings* serem úteis, eles apresentam limitações para sua aplicação direta em *sentence embedding*. Por exemplo, muitas vezes uma representação adequada de uma frase, ou de um documento inteiro, não é sempre obtida pela simples média dos vetores de palavras presentes nas frases. Literatura tem estudado formas mais complexas e efetivas de gerar essas representações [5] ou [7].

Cer *et al.* [5] discorreram sobre o significado presente nos vetores resultantes e como podem ser usados. Este trabalho descreveu que os vetores produzidos pelo modelo estão num espaço vetorial em que similaridade por cosseno entre vetores expressa o quão similares essas frases são. Por outro lado, Devlin *et al.* [7] descreveram que vetores resultantes podem ser usados para treinar redes neurais. Contudo, nenhuma função matemática é definido nesse espaço para se calcular similaridades, ou seja, eles fazem sentido para uma rede neural, porém não fazem sentido num espaço vetorial euclidiano, por exemplo.

Perone *et al.* [15] apresentaram e compararam diversas técnicas para se gerar *sentence embeddings*. Algumas dessas técnicas são baseadas em abordagens similares a Cer *et al.* [5].

Perone *et al.* [15] demonstraram que gerar *sentence embeddings* ainda é uma questão em aberto na literatura e não existe uma forma definitiva. Nossa investigação consiste em usar as *sentence embeddings* produzidas pela rede *USE* [5] (já pré treinada). Então, modificamos esses vetores durante o treinamento da nossa rede proposta, a *DeepSim* (cf. Seção 3). Em nossa abordagem, usando esse conceito de *sentence embeddings*, buscamos pelas frases mais similares a uma nova frase dada num

espaço vetorial, que contém os vetores que representam essas frases. Esse processo é a recuperação de informação do nosso sistema.

Nossa proposta explora *redes neurais siamesas* [16] que são geralmente usadas em problemas de classificação em o objetivo é aprender a maximizar uma certa função de similaridade para um par de entradas. Durante o processo de treinamento da rede, a função de similaridade corresponde a uma similaridade semântica entre as frases, já que a rede aprende a gerar representações robustas dos dados de entrada. É nesse ponto que esse tipo de rede ajuda no nosso problema, pois objetivamos que nossa rede neural, *DeepSim*, aprenda a identificar se pares perguntas são de fato similares e podem ter uma mesma resposta candidata.

Esse conceito de usar redes neurais para comparar a similaridade de objetos, pode ser usado para dado um novo objeto, comparar ele com cada objeto previamente conhecido e ranqueados de acordo com essa similaridade. Esse processo é a parte de ranqueamento do nosso sistema.

Entendemos que com a capacidade de aprender representações mais robustas, podemos alterar as representações dadas pela rede *USE* para que elas levem mais em consideração certas estruturas linguísticas que são mais relevantes para o nosso contexto. Além dessa aplicação, *redes neurais siamesas* são usadas quando o número de dados disponíveis é limitado através de técnicas como *Constrastive Loss* [3], *Triplet Loss* [3], *One Shot Learning* [3] entre outras.

Um dos problemas atuais em inteligência artificial é o (re)treinamento de modelos de *PLN* para várias línguas. Para tratar esse problema, podemos usar um *modelo multi linguístico*, que consiste de um modelo treinado para várias línguas ao mesmo tempo. Por exemplo, modelos desse tipo como *BERT* [7], *USE* [5] ou *XLM* [9]. Esses modelos mostraram que é possível treinar um modelo linguístico em mais de uma língua e igualmente demostraram diversos benefícios disso. Eles se tornaram modelos do estado da arte quando foram publicados. Nesta investigação, exploramos o modelo treinado em [5] para então, treinar uma rede em uma segunda língua visando classificar similaridade num contexto específico.

Esta pesquisa endereça igualmente o conceito de *Busca Semântica*, que consiste em buscar por um objeto usando o significado semântico desse objeto [2]. Por exemplo, buscar em um banco de dados de imagens como o *Google* usando um texto que expressa o sentido daquela imagem. Esse tipo de tarefa permite que usuários que não conhecem ou não tem acesso a identificadores únicos do sistema, consigam achar o que estão procurando e obtenham resultados que os mesmos não sabiam que estavam buscando. Por exemplo, um usuário pode buscar por um artigo numa revista e acabar descobrindo artigos similares à aquele que sejam ainda melhores. Tracz *et al.* [12] apresentou inúmeras técnicas sobre esse conceito e suas implementações.

A relação de *Busca Semântica* com nosso trabalho é que a entrada do nosso modelo é obtida através de uma busca semântica realizada através do banco de dados *Elasticsearch* (cf. Seção 3). Esse armazena todas as perguntas recebidas por uma loja

do *Mercado Livre*, separados pelo produto para o qual as perguntas foram realizadas. Em síntese, quando uma nova pergunta é feita pelo usuário no *Mercado Livre*, nosso sistema a envia para o *Elasticsearch* que faz uma *Busca Semântica* retornando um subconjunto das perguntas mais similares com a nova pergunta demandada (etapa de recuperação). Com base nesse resultado, nosso sistema escolhe a pergunta desse subconjunto mais similar (etapa de ranqueamento) a nova pergunta e reutiliza a resposta dela (cf. Seção 3).

2.2 Trabalhos Relacionados

Pesquisas aplicadas a tarefa de similaridade semântica é extensa [2]. Focamos nossa análise da literatura em trabalhos tratando similaridade entre textos e que usem redes neurais para tal tarefa.

Liu *et al.* [10] propuseram a rede *Robert*. O trabalho consistiu em avaliar o pré treino feito em Devlin *et al.* [7] e demonstrar que não apenas a rede *BERT* [7] foi sub treinada, mas também que fazer processo de *finetuning* nessa rede consegue igualar ou superar outros resultados. Nosso estudo conduz um procedimento similar, mas enfatiza conjunto de dados em outra língua.

Koroleva *et al.* [8] utilizaram diversos sistemas baseados na rede *BERT* [7] para treinar um classificador de similaridade que toma como entrada relatórios médicos. O sistema é capaz de identificar se esses relatórios são similares. De forma análoga, nosso trabalho considera textos como similares, somente se tratam do mesmo produto e ambos os textos tem a mesma intenção. Assim é preciso que a rede foque em certos termos das frases para dizer se são ou não similares. Realizamos igualmente uma anotação manual em um conjunto de dados para aprimorar os resultados. Nosso trabalho difere principalmente pela rede usada como base e no domínio do problema.

No contexto de comércio eletrônico, Zhang *et al.* [21] aprimoraram um motor de busca usando, dentre outras técnicas, a similaridade semântica entre buscas feitas por usuários. Esse trabalho se relaciona ao nosso, por ser aplicado no mesmo domínio e por usar técnicas similares para classificar textos. Ele se difere no sentido de utilizar técnicas distintas para se obter uma melhor efetividade nos resultados. Por exemplo, usa a intenção das *queries* para auxiliar na recuperação de informação; usa também a taxonomia das categorias dos produtos. Um dos motivos para

Não seguimos uma abordagem mais abrangente como feito em Zhang *et al.* [21] pois não temos um classificador de intenções confiável para essa tarefa. Em nosso contexto, a categorização dos itens na plataforma *Mercado Livre* é muitas vezes efetuada pelos lojistas. Encontramos classificações conflitante e problemáticas nesse contexto.

Tracz *et al.* [20] usaram a rede *BERT* de forma similar a que usamos a rede *USE* em nossa investigação. O objetivo deles foi classificar produtos e não perguntas como similares. Em nosso contexto de estudo, um dos motivos, dentre outros, para não usarmos a rede *BERT* no nosso sistema é que ela é mais computacionalmente exi-

gente que a rede *USE*. Adicionalmente, os vetores que representam textos produzidos pela rede *BERT* possuem maior dimensão, o que os tornam mais difíceis de serem guardados num banco de dados em larga escala.

3 *DeepFAQ*: Um sistema de respostas baseado em similaridade de sentenças

Denominamos o sistema que encapsula o *DeepSim* (nossa proposta) de *DeepFAQ*. Esse sistema está organizada em duas partes: *Buildtime* (cf. Subseção 3.1) e *Runtime* (cf. Subseção 3.2). A *Buildtime* consiste em operações que são executadas antes de o sistema realmente responder a uma pergunta; A parte de *Runtime* consiste em operações realizadas durante a execução do sistema para encontrar uma resposta para uma nova pergunta que chega ao sistema.

3.1 *Buildtime*

Antes que o sistema seja capaz de responder a uma pergunta recebida para uma determinada loja online em uma plataforma de *e-commerce*, é necessário gerar um espaço vetorial em que seja possível recuperar possíveis perguntas candidatas. Uma rede neural, que denominamos de *DeepSim*, classifica as questões sobre sua similaridade. Ela é treinada para que seja possível utilizá-la no *Runtime* (cf. Subseção 3.2). A Figura 1 apresenta os componentes e procedimentos envolvidos no *Buildtime*. As próximas subseções explicam cada componente em detalhes.

3.1.1 Recuperação e pré-processamento de perguntas e respostas

O primeiro passo para construir um espaço vetorial com possíveis perguntas candidatas é recuperar os pares pergunta-resposta de uma loja. Para isso construímos um *script* que acessa a *API* do *Mercado Livre* e recupera as perguntas de uma determinada loja. Obtém-se todos os pares de perguntas e respostas da história da loja desde que a loja aderiu à plataforma de comércio eletrônico *Mercado Livre*. Assim que obtivemos os pares de controle de qualidade da loja, um pré-processamento é executado neles como segue: as palavras de interrupção são removidas; os caracteres são convertidos para minúsculas e os acentos são removidos. O processo de remoção de palavrões das respostas é de grande importância, pois as respostas do atendimento ao cliente geralmente contêm saudações baseadas no horário do dia (*e.g.* bom dia, boa noite, *etc*), a assinatura do funcionário da loja, e às vezes o nome do cliente que enviou a pergunta.

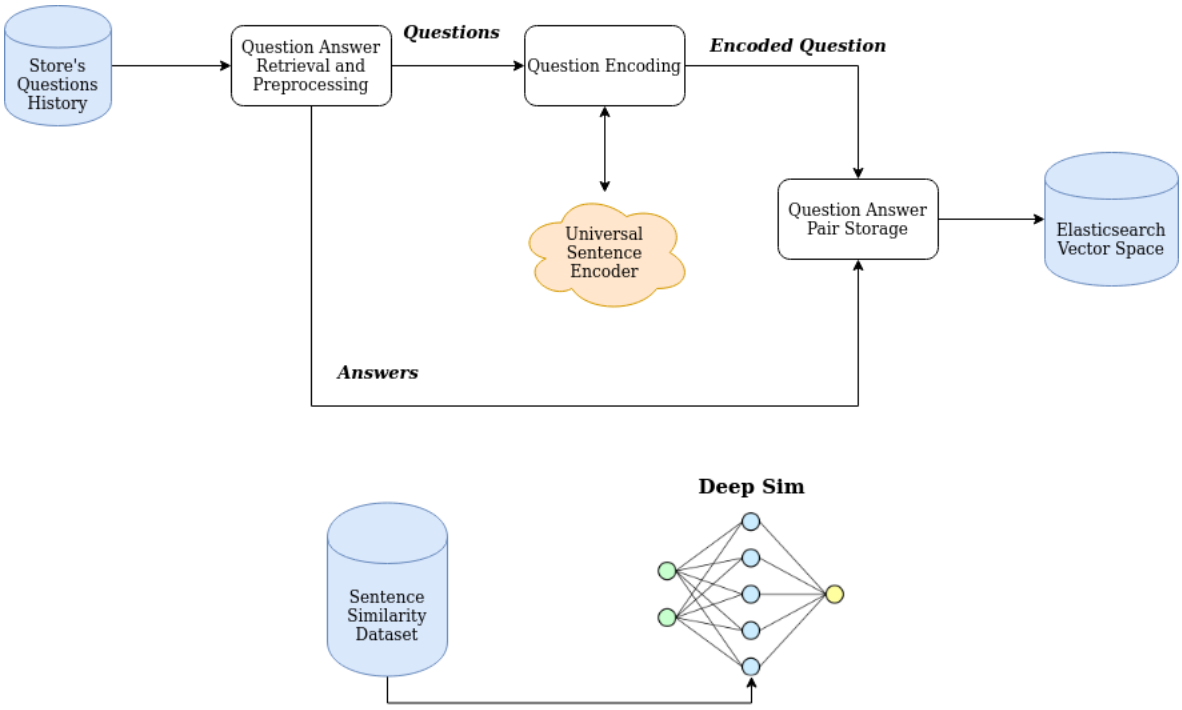


Figura 1: *Buildtime* do sistema

3.1.2 Codificação de perguntas

Uma vez que os pares de perguntas e respostas são limpos, as questões são codificadas usando o *Codificador de Sentença Universal* [5]. O processo de codificação consiste em enviar perguntas a um serviço remoto e obter a incorporação da frase como resposta. Nesse sentido, nossa solução é adequada para substituir o *USE* [5] por qualquer outro codificador que possa gerar uma incorporação de frase para a qual a semelhança de cosseno faça sentido.

Para o cálculo de similaridade de sentenças, exploramos a função de similaridade de cosseno θ (cf. Equação 1), em que a similaridade é o cosseno do ângulo entre dois vetores diferentes de zero. Em nosso caso de uso, cada vetor é uma questão codificada pelo *USE* [5].

Optamos por explorar esse codificador devido à sua simplicidade e ao fato de já ter sido treinado em tarefas de similaridade de texto. Nesse sentido, podemos usá-lo de forma direta e sem modificações, sem precisar ajustá-lo para nossa tarefa específica. Além disso, um grande benefício considerando que conjuntos de dados de similaridade de texto na língua Portuguesa são raros, ainda mais em se tratando do contexto de *e-commerce*.

3.1.3 Armazenamento de perguntas e respostas

Depois de obter os *embeddings* de frases, cada par de QA (um par *Question-Answer*) e seus *embeddings* de perguntas são enviados para nosso banco de dados de espaço vetorial implementado com *Elasticsearch*. Isso evita o armazenamento de perguntas repetidas para que a etapa de recuperação do candidato em *Runtime* não seja afetada (cf. Subseção **??**). Junto com o par de controle de qualidade e a incorporação da pergunta, o *id do produto* para o qual a pergunta foi feita também é armazenado.

3.1.4 Treinamento do *DeepSim*

Em paralelo ao processo de população do banco de dados de pares de perguntas e respostas, uma versão inicial da nossa rede neural *DeepSim*, foi treinada usando somente dados em português. Essa versão é atualmente usada em ambiente de produção pela *GoBots* e só é similar a rede treinada com dados em português que será apresentada mais adiante nesse trabalho.

Seção 4 apresenta como os hiper parâmetros dessa rede foram obtidos assim como o conjunto de dados usados para o treinamento.

3.2 *Runtime*

Figura 2 apresenta o sistema com os componentes envolvidos no tempo de execução da solução. O processo de responder a uma nova pergunta explora o banco de dados de espaço vetorial construído (em *Buildtime*) (cf. Subseção 3.1) com o histórico de perguntas e respostas para uma determinada loja online na plataforma de *e-commerce*. A solução desenvolvida é adequada para classificar um par de questões como semelhantes ou não. Os detalhes do funcionamento do sistema são explicados com mais detalhes nas subseções a seguir.

3.2.1 Pré-processamento e codificação da pergunta

O pré-processamento realizado na pergunta recebida é semelhante ao feito no *Buildtime*. A questão de entrada processada é então codificada com a rede *USE* e *embedding* da frase é obtida.

3.2.2 Recuperação de perguntas mais semelhantes

Com a pergunta codificada disponível, uma consulta é realizada no banco de dados que contém o espaço vetorial das perguntas anteriores existentes. Este procedimento recupera as k questões mais semelhantes usando a técnica de similaridade definida na Equação 1; k é um parâmetro no sistema. O conjunto com os candidatos em potencial

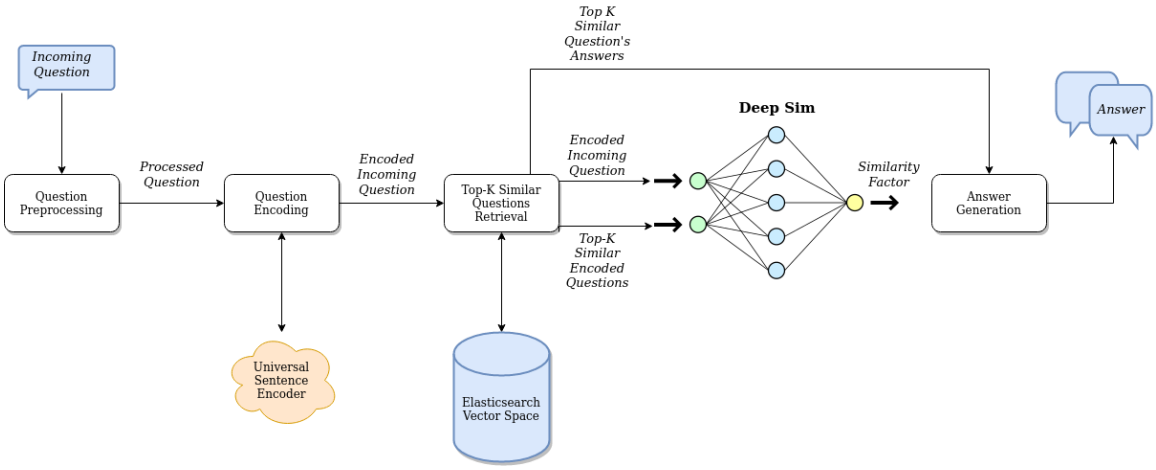


Figura 2: *DeepFAQ* – sistema implantado para responder às novas questões de clientes em uma loja de comércio eletrônico online

não apenas contém suas perguntas codificadas, mas seu par de controle de qualidade em linguagem natural. Na Equação 1, NQ é o vetor gerado pela codificação da *Nova Questão* que é recebida pelo sistema; Q é o vetor de uma questão já respondida pela loja em nosso espaço vetorial.

Elasticsearch não implementa nativamente uma maneira de pesquisar as k questões principais. Para tanto, usamos então o *AWS Elasticsearch Service* que é baseado em *Elasticsearch Open Distro* que acompanha o algoritmo *KNN* [6] pronto para usar neste ambiente.

$$\cos\theta = \frac{Q \cdot NQ}{\|Q\| \, \|NQ\|} \tag{1}$$

3.2.3 Classificação de perguntas candidatas por meio da rede *DeepSim*

Em algumas tarefas, apenas esta primeira recuperação seria suficiente para fornecer resultados ao usuário final. Em nosso caso, ainda precisamos de mais um passo para permitir responder às perguntas corretamente. A razão para isso é que os vetores fornecidos pelo modelo *USE* encapsulam a intenção de uma questão, mas falha em notar que o objeto da questão é diferente. Em nosso contexto, isso é crucial porque uma determinada pergunta sobre alguma especificação de um produto A pode ter uma resposta completamente diferente de um produto B. A Tabela 1 apresenta um exemplo disso; a pontuação de semelhança de cossenos para essas questões é alta e pode levar a um erro.

Para resolver isso, cada questão do conjunto dos k principais candidatos é usada como entrada para a rede *DeepSim*, em conjunto com as questões de entrada codifi-

Tabela 1: Exemplo de porque precisamos de um segundo sistema de classificação, Q é a nova questão, SQ é uma questão semelhante encontrada pelo *Elasticsearch* e CS é a semelhança de cossenos entre eles.

Q:	*Teria a cor da madeira no tom marron mais escuro?*
SQ:	*Teria a cor da madeira em preto ou embuia*
CS:	0.83

cadas. Isso permite usar a saída da rede para classificar cada elemento no conjunto dos k candidatos principais, associando-os a uma confiança, gerada pelo *DeepSim*, de ser semelhante à pergunta original.

3.2.4 Gerando uma resposta

Uma vez que os candidatos foram classificados, a solução seleciona o candidato no conjunto com a maior confiança dada pelo *DeepSim*. Se a probabilidade estiver acima de um certo limite (um parâmetro no sistema), a resposta do candidato é usada como resposta à pergunta recebida do cliente. Este limiar é definido dinamicamente para cada loja online na plataforma de *e-commerce*.

Na geração da resposta, o pós-processamento é realizado na resposta, como adicionar saudações e assinaturas da loja. Além disso, verifica se há *URLs* mortas na resposta, pois é comum vincular o usuário a outro produto nesses tipos de respostas. Se houver uma *URL* que não está mais disponível, a resposta é descartada para não frustrar o cliente.

3.2.5 Populando o banco de perguntas e respostas continuamente

Quando o sistema falha em responder a uma pergunta, um atendente humano responde à pergunta na plataforma como faria se não houvesse um *chatbot*. Quando o humano faz isso, implementamos um sistema que envia uma notificação para o *DeepFAQ*, e então, armazena essa nova pergunta respondida. Isso nos permite estar sempre atualizados com novos produtos e questões e, assim, responder a mais perguntas.

4 Avaliação Experimental

Esta seção detalha os experimentos realizados e os resultados obtidos.

4.1 Preparação dos conjuntos de dados

Primeiramente, coletamos, tratamos e separamos os dados usados. A coleta dos dados considerou diretamente perguntas feitas por clientes na língua Espanhola na plataforma *Mercado Livre*. Essas perguntas foram anonimizadas e separadas em pares usando o algoritmo de clusterização *KNN* [6]. Para cada pergunta coletada, codificamos usando a rede *USE*. Em seguida, a pergunta foi passada para o algoritmo *KNN* para se agrupar perguntas mais próximas em pares. Montados os pares de perguntas, cada uma delas foi anotada por um humano, fluente na língua Espanhola, considerando uma *tag* de similar ou não similar. Esse conjunto possui 3000 pares de perguntas balanceado, ou seja, 1500 similares e 1500 não similares.

Os pares de perguntas em português foram obtidos através do sistema usado em produção pela *GoBots*. Quando esse sistema tenta encontrar uma pergunta similar a uma nova pergunta feita por um cliente, ele guarda num banco de dados a pergunta similar encontrada (caso exista) e os candidatos (não similares) encontrados durante o processo, formando pares com a nova pergunta recebida. Usamos esses pares para montar o conjunto de dados na língua Portuguesa, que em seguida foi validado por humanos conferindo se o sistema acertou em marcar o par como similar ou não similar. Esse conjunto conta com 20000 pares de perguntas balanceadas, ou seja, metade similar e metade não similar.

O conjunto de dados em inglês foi obtido de forma bem diferente dos demais. Como a *GoBots* atualmente não trabalha em países cuja língua nativa é o inglês, não existem perguntas de clientes feitas em inglês, logo foi preciso buscar um outro conjunto de dados que fizesse sentido para o nosso contexto. O *dataset* escolhido foi o *Quora Question Pairs* [18], que é um conjunto de dados composto por pares de perguntas retiradas do website *Quora* [1]. De forma resumida, nesse site usuários fazem perguntas de qualquer assunto e outros usuários as respondem.

O *Quora Question Pairs* contém mais de 250000 pares de perguntas, então selecionamos aleatoriamente 10000 perguntas classificadas como similares e 10000 não similares, formando assim um conjunto com 20000 perguntas, mesmo tamanho do nosso conjunto de dados de sentenças na língua Portuguesa.

Os 3 conjuntos de dados foram tratados da seguinte forma: todas as letras foram passadas para a forma minuscula. Nomes, e-mails e qualquer outro dado que possa identificar algum usuário foram removidos.

A partir desses 3 conjuntos, um total de 6 conjuntos foram formados combinando os 3 originais. O primeiro é um conjunto de dados usado para testes nos experimentos que contém 400 perguntas em espanhol. O segundo é um conjunto com 2600 pares de perguntas em espanhol. O terceiro é composto por 2600 perguntas em espanhol e 2600 em português, ou seja, balanceado. O quarto por 2600 perguntas em espanhol e 20000 em português. O quinto por 2600 perguntas em espanhol e 2600 em inglês,

[1] https://quora.com

ou seja, balanceado. O sexto, e último, por 2600 perguntas em espanhol e 20000 em inglês.

Com exceção do conjunto de testes (em Espanhol), todos os outros 5 conjuntos foram divididos em subconjuntos com tamanhos indo de 400 pares de perguntas, até o tamanho total do conjunto variando em 400 pares de perguntas entre cada subconjunto [2].

Em cada um dos subconjuntos com somente uma língua, i.e., dados somente em espanhol ou português ou inglês, usamos 10% do seu tamanho como conjunto de validação durante o processo de treinamento, sendo que esses 10% foram escolhidos de forma aleatória entre cada treino. Fizemos essa escolha de forma sempre aleatória para garantir que uma boa ou má escolha de certos dados não interfira nos resultados dos experimentos.

Nos subconjuntos com duas línguas misturadas, i.e., conjunto de dados com espanhol e português ou espanhol e inglês, a escolha de dados para o conjunto de validação foi feita de duas formas: a primeira, nos casos de conjunto de dados balanceados [3]usamos 10% dos dados de cada língua, e.g., adicionamos 400 questões em espanhol e 400 em português e usamos 40 questões em espanhol e 40 em português como conjunto de validação. A segunda, nos casos em que os dados não estão balanceados, escolhemos 10% do tamanho total do conjunto menor, i.e., das perguntas em espanhol, e o mesmo número de questões do conjunto maior, i.e., português ou inglês. Em ambas as formas a escolha dos dados para o conjunto de validação foi feita de forma aleatória.

4.2 Experimentos

O objetivo geral de nossos experimentos é averiguar o quanto o treino da rede em línguas diferentes ajuda uma outra língua (efetividade que uma rede pode atingir usando uma rede treinada em uma certa língua com dados escritos em outra língua); igualmente quanto o volume de dados no conjunto de treinamento impacta na nossa tarefa.

Em todos os experimentos testamos a rede neural proposta com um mesmo conjunto de teste em espanhol e comparamos a acurácia e a perda alcançadas em cada experimento pela rede. Equação 2 define a medida de acurácia enquanto a Equação 3 define a medida de perda.

[2]E.g. o conjunto de dados com 2600 pares de perguntas em espanhol foi dividido em subconjuntos de tamanho 400, 800, 1200, ..., 2600

[3]Aqui nos referimos a conjuntos de dados balanceados como conjuntos com o mesmo número de dados em cada uma das línguas

$$Accur = \frac{TP + TN}{T} \qquad (2) \qquad\qquad Loss = \frac{1}{n}\sum(Y_i - \hat{Y})^2 \qquad (3)$$

Na equação de acurácia 2, nelas TP significa *True Positives*, TN *True Negatives*, T significa *Total*. Na equação de perda 3, n significa o total de exemplos, Y_i é o valor de saída da rede para o exemplo i, $\hat{Y_i}$ é o valor médio da saída da rede.

Em cada experimento o treino da rede será rodado 5 vezes com o mesmo conjunto para mitigar o efeito que uma inicialização inadequada dos pesos da rede tem sobre sua efetividade. Após cada treino, usamos nosso conjunto de testes em espanhol para determinar a acurácia final. Usamos sempre o conjunto em espanhol por duas razões principais: a primeira é que usando um conjunto de testes somente em uma língua podemos ver como outras línguas influenciam na performance na língua do conjunto de testes; o segundo é que a *GoBots* está expandindo para países Sul-Americanos que falam espanhol, então avaliar o desempenho nessa língua é importante.

Os resultados apresentados são a média simples do resultado obtido em cada um dos 5 treinos. Os treinos duraram no máximo 100 épocas e foram parados caso não houvesse melhoria de ao menos 10^{-4} na acurácia de validação da rede após 3 épocas.

Todos os experimentos usaram os mesmos hiper parâmetros, já que nosso foco não foi avaliar a interferência deles na efetividade da rede, mas a interferência de dados (sentenças em linguagem natural) de línguas diferentes e as quantidades. Para esse fim, usamos o algoritmo *Random Search*, mais especificamente sua implementação no *framework Keras* [13], para encontrar os melhores hiper parâmetros para o modelo usando todos os dados em espanhol. Os hiper parâmetros encontrados inicialmente e usados em todos os testes realizados foram os seguintes: pre_units: 0 pos_units: 1024 act: relu drop: 0.05 pre_num_layers: 0 pos_num_layers: 2 learning_rate: 0.001.

A tabela 4.2 apresenta uma síntese sobre os experimentos conduzidos descrevendo as questões de pesquisa que esperamos responder com eles. Na coluna *Resultados*, são mostrados a performance da rede no conjunto de validação e no teste obtidos com o uso todos os dados disponíveis em cada experimento, e.g. no experimento 1, os valores de *Validation Accur* e *Test Accur* foram obtidos com a rede treinada com os 20000 pares de pergunta em português.

Exp.	Descrição	Objetivos	Resultados
1	Usa um conjunto de dados com sentenças na língua Portuguesa, anotado e revisado manualmente; com 20.000 pares de perguntas obtidas de clientes reais, cuja língua nativa é o português em um ambiente de produção. Treinar uma rede com esse conjunto de dados e testar no conjunto de testes na língua Espanhola formado por perguntas de clientes reais, porém com pares criados através de um método de *clusterização*; cada um desses pares foram revisados manualmente para validar sua similaridade. O tamanho do conjunto de dados usado no treino será variado entre cada um dos treinos, para avaliarmos o impacto do volume de dados na efetividade da rede.	Determinar a efetividade de um modelo treinado em português,com um volume grande de questões anotadas. Entender até que ponto é possível usar esse modelo (treinando com sentenças na língua Portuguesa) para responder perguntas em espanhol, eliminando a necessidade de anotar perguntas em espanhol o que muitas vezes é demorado e caro. Objetivamos igualmente averiguar a quantidade de dados necessária para esse fim.	Validation Accur: 0.7800, Validation Loss: 1.303, Test Accur: 0.7346, Test Loss: 0.9655
2	De forma similar ao experimento 1, treinamos uma rede usando somente sentenças na língua Espanhola. Esses dados foram cerca de 3.000 pares formados através de um método de *clusterização* em perguntas feitas por clientes reais cuja língua nativa é o espanhol; esses dados foram manualmente revisados por humanos para validar a similaridade dos pares de questões. Variamos o tamanho do conjunto de dados entre os treinos para investigar a quantidade de dados necessária para se obter um resultado adequado esperado da rede.	Determinar se o uso de pares de sentenças em espanhol pode melhorar significativamente a efetividade obtida com relação aos resultados obtidos com o treino em português. Visamos analisar se os custos da anotação desses dados se justifica e responder o quanto de dados é necessário anotar para isso.	Validation Accur: 0.8202, Validation Loss: 0.4992, Test Accur: 0.8091
3	Misturamos os conjuntos de dados em espanhol com português de forma balanceada, ou seja, com mesmo número de perguntas em português e espanhol; e de forma desbalanceada, usando mais dados em português. Então repetimos a variação de volume de dados entre cada treino monitorando o resultado da mistura de dados na efetividade da rede.	Determinar o impacto que a mistura de línguas tem na rede. Responder se o aumento do volume de dados em outra língua pode ajudar na efetividade da rede. Determinar o impacto que o desbalanceamento entre essas línguas pode ter na qualidade dos resultados da rede.	Validation 1 Accur: 0.7635, Validation 1 Loss: 0.7915, Test 1 Accur: 0.7961, Test 1 Loss: 0.5445, Validation 2 Accur: 0.0.7659, Validation 2 Loss: 0.6094, Test 2 Accur: 0.8494, Test 2 Loss: 0.4360
4	Misturar os conjuntos em espanhol com inglês de forma balanceada, ou seja, com mesmo número de perguntas em inglês e espanhol e de forma desbalanceada, usando mais dados em inglês. Esse experimento é similar ao experimento 2 e segue os mesmos procedimentos descritos no mesmo, porém usando a língua inglesa ao invés do português	Determinar se os resultados do experimento 1 também são válidos para outras línguas além do português, principalmente para línguas cuja origem é diferente do espanhol, cuja a origem é o latim	Validation Accur: 0.7703, Validation Loss: 0.6577, Test Accur: 0.6828, Test Loss: 1.3344
5	Misturar os conjuntos em espanhol com inglês de forma balanceada, ou seja, com mesmo número de perguntas em inglês e espanhol e de forma desbalanceda, usando mais dados em inglês. Esse experimento é similar ao experimento 2 e segue os mesmos procedimentos descritos no mesmo, porém usando a língua inglesa ao invés do português	De forma similar ao experimento 3, objetivos avaliar os resultados obtidos nele, para uma língua cuja origem não seja o latim; e então dizer se a escolha da língua pode ser determinante para se expandir o conjunto de dados com outra língua ou não	Validation 1 Accur: 0.7648, Validation 1 Loss: 0.8274, Test 1 Accur: 0.8285, Test 1 Loss: 0.4727, Validation 2 Accur: 0.7945, Validation 2 Loss: 0.7203, Test 2 Accur: 0.8479, Test 2 Loss: 0.6071

O nosso primeiro experimento serve para termos um parâmetro de comparação para os nossos próximos experimentos (*baseline*). Nele, treinamos uma rede usando somente o conjunto de dados em português, primeiro com o número reduzido de perguntas; e segundo na mesma versão que é atualmente usado em produção, ou seja com todas as perguntas, treinaremos ela com os subconjuntos. Com isso podemos primeiro mensurar se conseguimos a mesma efetividade na rede tanto treinando com um conjunto de dados em português quanto em espanhol. Isso nos permite analisar se a nossa rede treinada com dados em espanhol (experimento 2) está próxima de um resultado que possa ser usado em produção. Para testar as redes treinadas com dados diferentes, em todos os experimentos, usamos o mesmo conjunto de testes em espanhol com 400 pares de perguntas.

Nosso segundo experimento consistiu em treinar a rede usando somente o nosso conjunto de dados em espanhol, e analisar o desempenho da rede para uma língua específica, ou seja a rede terá seus pesos reiniciados e retreinados do zero. Ainda dentro do segundo experimento, vamos realizar o procedimento de quebra e treino com subconjuntos com o conjunto de dados em espanhol usando os hiper parâmetros encontrados, verificando assim o impacto que o tamanho do conjunto de dados tem no modelo.

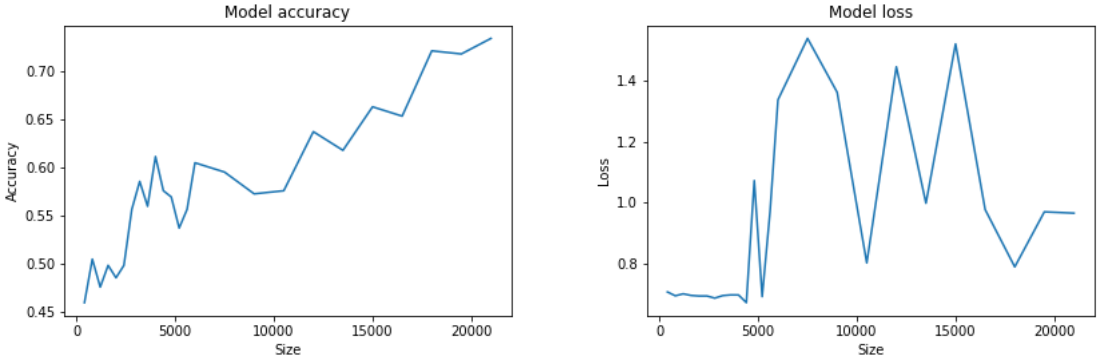
No terceiro experimento, misturamos o conjunto de dados em espanhol com o conjunto em português reduzido e com o conjunto de produção. Visamos comparar o impacto que a mistura de línguas tem na efetividade da rede (usando o conjunto de teste na língua Espanhola), e o impacto que um *dataset* multilinguístico desbalanceado, ou seja, com mais perguntas em uma língua do que em outra, tem. O conjunto de validação usado durante a busca por hiper parâmetros assim como durante o treino com sub conjuntos de dados, foi formado sendo 50% de dados em português e 50% de dados em espanhol, ambos escolhidos aleatoriamente.

Em nosso quarto experimento, prosseguimos de forma similar ao primeiro experimento, porém ao invés de usar dados em português, usamos dados da língua Inglesa. No quarto experimento usamos 20.000 dados em inglês de forma similar ao primeiro experimento. No quinto experimento usamos variamos o tamanho do conjunto de dados entre 800 e 22600 misturando com os nossos dados em espanhol de forma a similar ao terceiro experimento. O tamanho do conjunto em inglês é ao do conjunto de dados em português (primeiro experimento). O objetivo desse experimento foi analisar se os resultados obtidos no primeiro e no terceiro experimento foram causados, mesmo que parcialmente, pela similaridade entre a língua Portuguesa e a Espanhola, já que ambas as línguas são descendentes do latim. O inglês por ter origem germânica, pode nos ajudar a responder essa questão.

4.3 Experimento 1

No primeiro experimento, treinamos a rede *DeepSim* usando somente sentenças na língua Portuguesa, variando a quantidade de dados desse treino. Testamos essa

rede com o conjunto de dados em espanhol. Figura 3 apresenta os resultados desse teste. Conseguimos obter resultados melhores a medida que adicionamos mais dados. Obtivemos uma acurácia de 0,74 no conjunto de testes usando 18.000 pares de perguntas (18.000 como treino, 2.000 como conjunto de validação). No conjunto de validação em português, obtivemos 0,78.



(a) Acurácia de teste dado um certo tamanho de dados

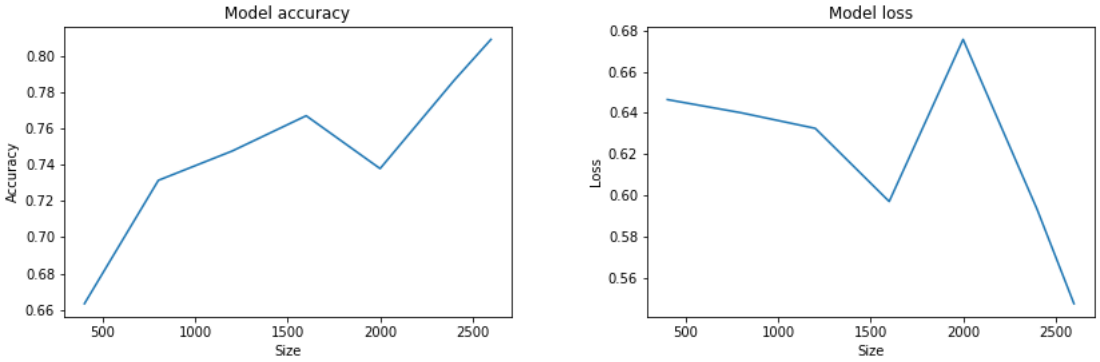
(b) Perda de teste dado um certo tamanho de dados

Figura 3: Resultados do experimento 1 (precisão e perda do treinamento). Treinamento da rede apenas com dados em português (treinado com vários tamanhos de dados) e testada com o conjunto de dados em espanhol.

Nesse primeiro experimento em que a rede foi treinada somente em português, o aumento no volume de dados permite a rede a extrapolar melhor, e atingir uma acurácia cada vez mais alta a medida que aumentamos os dados. Em todas as vezes em que repetimos esse experimento, a rede melhorou a sua efetividade a medida em que aumentamos o volume de dados para o treinamento; como não conseguimos achar um platô, não podemos dizer até que ponto um aumento no volume de dados ajudaria a rede de forma significativa. A perda porém se mostrou bastante instável o que indica que ao adicionarmos novos elementos ao treino, a rede perde confiança. Acreditamos que isso seja causado por problemas na qualidade da anotação desses dados.

4.4 Experimento 2

No Segundo experimento, replicamos os procedimentos adotados no primeiro, porém dessa vez usando um conjunto de dados em espanhol para treino; e dados em espanhol para teste. Variamos o tamanho do conjunto de dados usado para o treinamento. Chegamos em resultados similares ao do primeiro experimento. Constatamos o efeito que o tamanho do conjunto de dados tem no desempenho do modelo. Chegamos numa acurácia mais alta de 0,81 nesse experimento usando 2.600 pares de perguntas, contra 0,74 do primeiro usando 18.000 pares.



(a) Acurácia de teste dado um certo tamanho de dados

(b) Perda de teste dado um certo tamanho de dados

Figura 4: Resultados do experimento 2 (precisão e perda do treinamento). Treinamento apenas com dados em espanhol (treinado com vários tamanhos de dados). Teste com dados em espanhol.

Resultados do experimento 2 demonstra que usando 2.600 dados (pares de questões) em espanhol, a rede conseguiu chegar em 0,81 de acurácia e uma perda de 0,55. Esse valor de perda nos indica que não só a rede está acertando, mas está conseguindo fazer previsões com confiança alta na similaridade das frases.

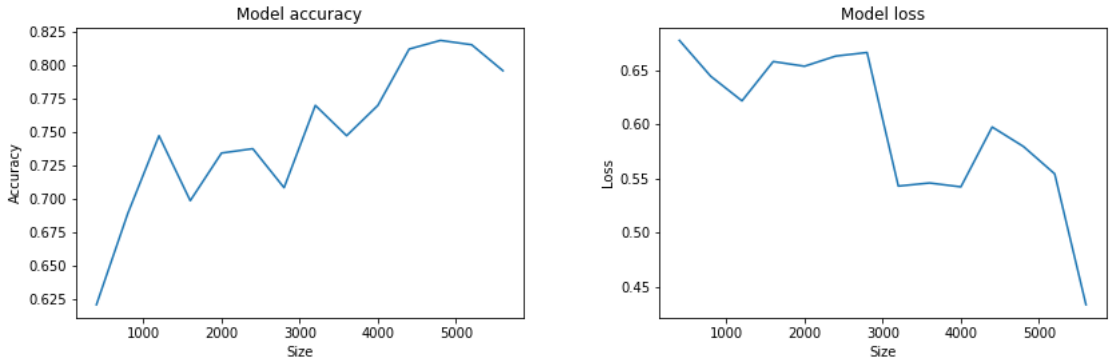
No experimento 2, observamos uma tendência de subida a medida que o volume de dados aumenta, sem conseguirmos enxergar uma tendência de estabilização (similar ao que observamos no experimento 1). No experimento 2 em que a rede foi treinada somente em espanhol, vemos um comportamento similar ao experimento 1, novamente não alcançamos um platô e claramente vemos que ocorre uma melhora a medida que adicionamos mais dados ao treinamento. No experimento 2, porém, não vemos uma instabilidade tão acentuada na perda, o que indica uma melhor qualidade nas anotações.

4.5 Experimento 3

No terceiro experimento, organizamos os resultados em duas partes. A Figura 5 apresenta os resultados que obtivemos quando treinamos nossa rede com até 2.600 dados em espanhol e 2.600 dados em português. Ou seja, o treinamento foi feito com um conjunto de dados balanceado. O teste foi feito com o conjunto de testes em espanhol. O tamanho do conjunto de dados também foi variado e para todos os tamanhos o número de perguntas em português e espanhol foi o mesmo.

A Segunda parte consistiu em um treino efetuado com até 2600 pares em espanhol e 20.000 pares de questões em português, variando o conjunto de dados. Ou seja, o treino foi feito de forma desbalanceada, com um número significativamente maior de

perguntas em português do que em espanhol. O teste foi conduzido com o conjunto de dados em espanhol.



(a) Acurácia de teste dado um certo tamanho de dados

(b) Perda de teste dado um certo tamanho de dados

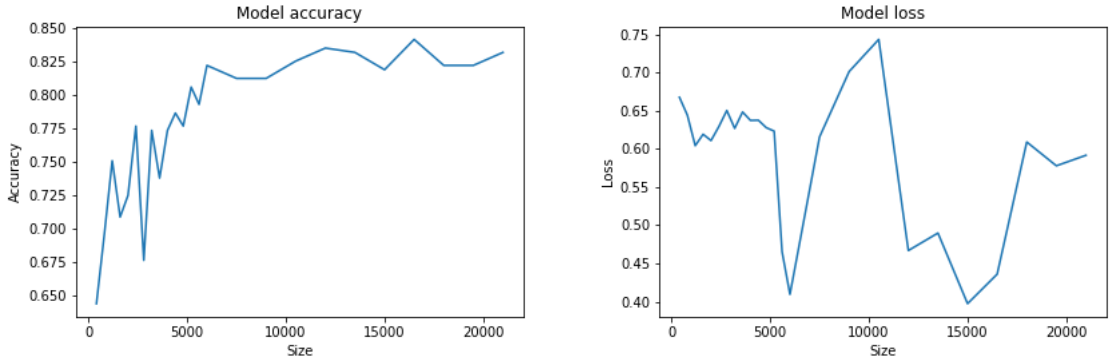
Figura 5: Resultados do experimento 3 parte 1 (precisão e perda do treinamento). Treinamento efetuado com dados em espanhol e português (balanceado) com vários tamanhos de dados. Teste efetuado com dados em espanhol.

A Figura 5 demonstra que a mistura dos dados em português e espanhol alcançou uma acurácia de 0,81, com cerca de 5.000 pares de perguntas no conjunto de testes em espanhol, enquanto a acurácia do conjunto de validação, que usa dados em português e espanhol (10% de cada um), foi de 0,76. Todos os pontos da Figura 5 foram obtidos usando o mesmo conjunto de dados em espanhol.

A Figura 6 apresenta os resultados da segunda parte do experimento 3. Obtivemos resultados similares ao da primeira parte, em que conseguimos uma acurácia de testes no conjunto em espanhol de 0,85; superior a encontrada na validação 0,77. Notamos também que depois de um certo tamanho do conjunto de dados a curva de acurácia se mantém razoavelmente estável até o final do experimento. A curva de perda oscila consideravelmente, de forma similar a curva vista no gráfico de perda do primeiro experimento (cf. Figura 3), em que somente dados em português foram usados no treino e o conjunto de testes era o mesmo.

Os resultados em ambas as partes do experimento 3 são próximos dos achados encontrados no experimento 2 em que a rede foi treinada somente com dados em espanhol.

Na parte 2 desse experimento, observamos que a melhora ocorre até certo ponto. A partir de um conjunto de dados de tamanho 6.000, ponto em que todos os dados em espanhol são incluídos no treinamento, vemos que a acurácia obtida pela rede fica estável e não apresenta melhoras significativas. Isso mostra que o simples aumento no volume de dados não necessariamente representa uma melhora, já que junto com o ganho de acurácia observado vem junto com uma grande variação na perda calculada.



(a) Acurácia de teste dado um certo tamanho de dados

(b) Perda de teste dado um certo tamanho de dados

Figura 6: Resultados do experimento 3 parte 2 (precisão e perda do treinamento). Treinamento efetuado com dados em espanhol e português (desbalanceado) com vários tamanhos de dados. Teste efetuado com dados em espanhol.

No experimento 3 parte 2 a acurácia obtida foi próxima (ligeiramente superior) ao do experimento 2. Outro ponto que corrobora com essa análise é que a acurácia alcançada no conjunto de validação do experimento 3 (0.77) foi próxima a acurácia de validação obtida no primeiro experimento (0.78), então da mesma forma que adicionar dados em português, não atrapalhou a rede em responder perguntas em espanhol, o inverso também parece ser verdadeiro.

Note que esse aumento na perda, não se verifica nos casos balanceados[4]. Neles conseguimos alguma melhora na rede, cerca de 2% de melhora, e diminuímos também a perda que foi de 0.55 no experimento 2, passa para 0.43 no experimento 3 parte 1 e 0.47 no experimento 5 parte 1.

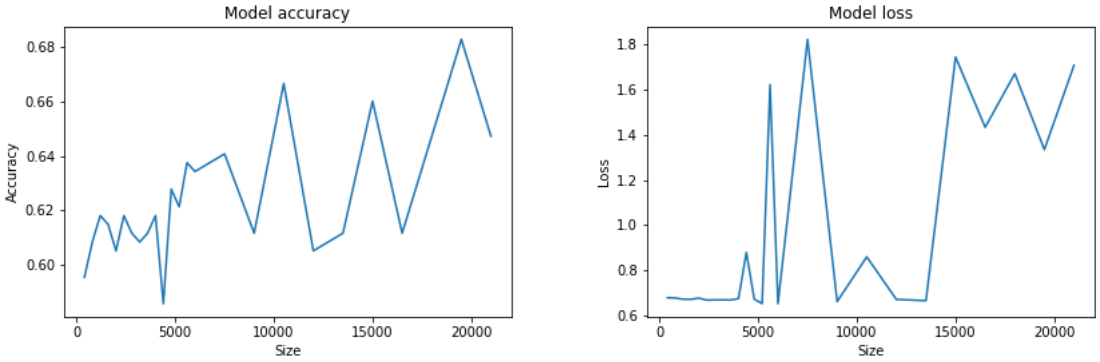
4.6 Experimento 4

No quarto experimento, repetimos o primeiro, porém usando dados em inglês e testando em espanhol.

A Figura 7 demonstra que com cerca de 20.000 pares de questões a acurácia alcançada foi de 0.68, abaixo da alcançada no experimento 1 que foi de 0.75, a perda também acabou sendo consideravelmente mais alta, sendo 1.34 no experimento 4 e 0.96 no experimento 1. A acurácia oscila ao adicionarmos novos dados, e fica mais difícil de se observar uma clara melhora, conforme foi observado nos resultados do experimento 1 (cf. Figura 3).

No experimento 4, a efetividade da rede é consideravelmente pior que a obtida no experimento 1, indicando que o fato de a língua Portuguesa e Espanhola serem

[4]Experimento 3, parte 1; e Experimento 5, parte 1;



(a) Acurácia de teste dado um certo tamanho de dados

(b) Perda de teste dado um certo tamanho de dados

Figura 7: Resultados do experimento 4 (precisão e perda do treinamento). Treinamento efetuado com dados somente em inglês (treinado com vários tamanhos de dados) e testado em espanhol.

mais próximas, auxilia a rede a aprender e leva a um resultado mais aceitável ao contrário do que acontece no inglês. Um ponto importante sobre essa análise, é que o conjunto de dados em inglês é composto por perguntas de domínio aberto. Além de diferir na língua, os dados diferem no assunto e contexto que tratam. Isso pode ter tido influência nesse resultado. Como não temos um conjunto de dados similar em português a nossa disposição[5], não é possível responder o quanto isso afeta exatamente.

4.7 Experimento 5

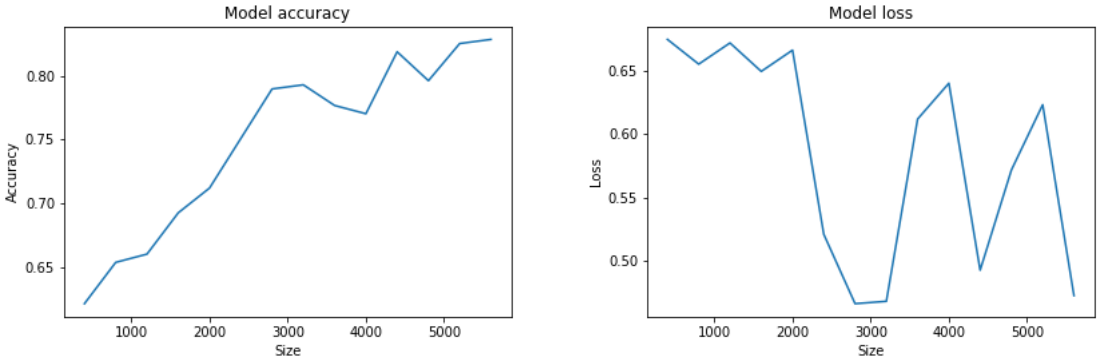
O quinto experimento foi igualmente organizado em duas partes. Prosseguimos de forma similar ao experimento 3, mas usamos dados em inglês ao invés de dados em português mantendo o tamanho do conjunto de dados e a variação de tamanho a cada teste.

Na primeira parte do experimento 5 (cf. Figura 9), alcançamos resultados similares aos obtidos na primeira parte do experimento 3, sendo uma acurácia de teste 0.83 contra 0.81 do experimento 3, e a perda se comportou de maneira similar.

Na segunda parte do experimento 5 (cf. Figura 9), obtivemos uma acurácia de 0.85, sendo ligeiramente melhor do que a obtida na segunda parte do experimento 3, as oscilações na perda se mantiveram.

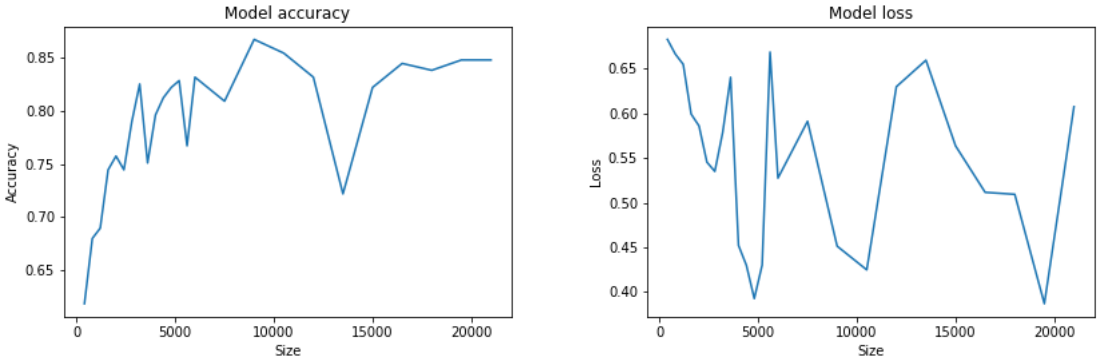
Observamos que no experimento 5 a rede conseguiu uma efetividade similar no

[5]o único disponível seria uma versão traduzida de forma automática do *Quora Question Pairs* feita pela *GoBots*. Isso introduziria um novo problema que é a qualidade da tradução gerada e isso está fora do escopo deste trabalho.



(a) Acurácia de teste dado um certo tamanho de dados (b) Perda de teste dado um certo tamanho de dados

Figura 8: Resultados do experimento 5 – parte 1 (precisão e perda do treinamento). Treinamento com dados em espanhol e inglês (balanceado) (treinado com vários tamanhos de dados). Conjunto de teste em espanhol.



(a) Acurácia de teste dado um certo tamanho de dados (b) Perda de teste dado um certo tamanho de dados

Figura 9: Resultados do experimento 5 – parte 2 (precisão e perda do treinamento). Treinamento com dados em espanhol e inglês (desbalanceado) (treinado com vários tamanhos de dados). Conjunto de teste em espanhol.

conjunto de testes ao experimento 3. Isso indica que os dados em inglês, apesar de suas diferenças, não atrapalharam a rede na tarefa de classificar perguntas em espanhol. Uma vez que o modelo tem acesso a todas as perguntas em espanhol, a acurácia e perda ficam estáveis e próximas aos resultados encontrados no experimento 2.

Notamos que no experimento 3 – parte 2 e experimento 5 – parte 2 (usando conjuntos de dados desbalanceados), a oscilação na perda, principalmente no experimento 5 é bem maior do que usando conjuntos de dados balanceados. Ao mesmo tempo que temos essa oscilação, não obtivemos resultados muito melhores devido ao maior número de dados, mantendo a acurácia sempre próxima dos 0.81 alcançados no experimento 2, chegando a no máximo 0.85 em um melhor caso. A taxa de perda ficou aproximada a 0.65 contra 0.55 no experimento 2.

Isso indica portanto, que a mistura de línguas no conjunto de dados não interfere com os resultados das outras línguas além da própria. Indica igualmente que a mistura de dados, visando aumentar o volume total de dados, apesar de melhorar a acurácia, acabamos com uma perda maior nos casos desbalanceados, o que afeta a confiança da rede nas suas predições e compromete o uso dos valores de saída da rede como um parâmetro válido de quão similar ou não duas perguntas são.

Resultados obtidos sinalizam que usar dados de outra língua ajude de forma moderada no desempenho da rede, mas essa melhora está limitada ao tamanho original do seu conjunto de dados. O aumento irrestrito dos dados gera aumento da perda da rede, o que pode levar a resultados piores, principalmente se usarmos a saída da rede como uma garantia de confiança nas respostas encontradas.

5 Discussão

O primeiro ponto de nossa discussão se refere a efetividade da rede treinada somente com dados em espanhol e com dados somente em português. O motivo dessa análise é que num ambiente de produção, caso a efetividade do modelo treinado em português seja suficientemente adequado, poderíamos usar ele diretamente em outras línguas (como o espanhol), sem ter que coletar, tratar e classificar dados em espanhol, ou ao menos postergar essa tarefa. Os resultados de nossos experimentos demonstraram que mesmo um conjunto de dados consideravelmente pequeno em espanhol, pode melhorar consideravelmente a efetividade da rede produzida. Para certos casos de perguntas, em que a acurácia não seja tão vital para o cliente, a rede em português mostrou ser útil caso não exista um conjunto de dados pronto para a outra língua.

O mesmo não se observou no experimento 4 em que a rede teve um desempenho significativamente pior no conjunto de teste, ou seja, tendo um conjunto de dados em inglês em muitos casos (como o da *GoBots*) não possibilita efetuar uma reutilização direta. Isso nos indica que a escolha das línguas em que se quer usar para esse tipo de solução desenvolve uma papel relevante.

Os resultados demonstraram que a rede apresenta uma melhora significativa na acurácia alcançada no conjunto de testes a medida que adicionamos mais dados,

indicando assim que anotar mais dados pode melhorar o desempenho da rede, o que era esperado. Nos experimentos 3 e 5, em particular, observamos uma melhora, porém depois de um certo volume de dados, a acurácia se estabilizou. Acreditamos que um dos fatores principais que causou isso, tenha sido a versão da rede *encoder* usada para transformar as frases em vetores. Usamos a versão multilinguística do *USE* [5] que é previamente treinada em 18 línguas diferentes com várias tarefas de PLN, de forma que o vetor gerado na saída não só represente similaridade entre frases de uma língua, mas entre frases de diversas línguas. Os experimentos 3 e 5 demonstraram que conseguimos treinar a rede em uma tarefa específica, transferindo conhecimento da rede *encoder*.

Temos que considerar que a taxa de perda variou muito em todos os experimentos. O comportamento esperado seria que a medida em que adicionamos mais dados a rede, a taxa de perda diminuísse. No entanto, existem duas explicações para ela oscilar bastante, principalmente em experimentos envolvendo mais dados: 1) a medida que adicionamos dados em outra língua ao nosso treino, fazemos a rede prever com menos certeza. Ou seja, na Equação 3, o termo $(Y_i - \hat{Y}_i)$ se torna maior, pois a previsão feita pela rede se torna mais próxima de 0.5. Isso pode ser causado por diferenças estruturais das línguas, falsos cognatos entre diversos fatores; 2) é possível que a qualidade dos dados em português e inglês não seja ótima, possivelmente sendo incoerentes internamente. Isso pode causar essa oscilação na perda. Por internamente incoerentes, queremos dizer que certas frases similares dentro do conjunto podem hora ser classificadas como similares e hora serem classificadas como não similares, fazendo a rede perder a certeza nas predições efetuadas.

Os experimentos 4 e 5 visaram analisar se o uso de uma língua mais distante em termos de estrutura e gramática pode prejudicar o desempenho da rede de alguma forma. Isso nos ajuda a decidir entre misturar ou não conjuntos de dados em certas línguas com outras. Observamos no experimento 4, que o uso do conjunto de dados com sentenças na língua Inglesa apresenta uma efetividade consideravelmente pior nos testes em espanhol. Então, dificilmente poderia ser usado sozinho em um ambiente de produção para responder perguntas em espanhol. Por outro lado, o experimento 5 demonstrou que misturar dados em espanhol com inglês melhorou os resultados da rede de forma significativa. Então desde que mantivermos os dados em línguas diferentes com volumes parecidos no conjunto de dados, podemos sim melhorar os resultados da rede, sem comprometer a confiança das suas predições.

A mistura de línguas num conjunto de dados, para a tarefa em questão, não atrapalha a efetividade da rede. Com base em nossos achados, acreditamos que aumentar o volume de dados, mesmo que em outra língua gera uma melhora moderada na acurácia mesmo em outra língua. Essa melhora só ocorre até certo ponto, depois disso, essa melhora vem ao custo de um aumento significativo na perda. Nossos resultados indicaram que o uso de línguas de mesma origem ou de origem diferente não interferem na efetividade da rede em uma língua específica, podemos misturar espanhol com inglês ou português sem comprometer os resultados. Mais relevante do

que a mistura de línguas no conjunto de dados, é a qualidade da anotação, sendo está mais crucial para o desempenho da rede na tarefa.

Verificamos que o maior uso de dados melhora consideravelmente o desempenho da rede em uma língua, desde que esses dados adicionados a mais sejam nessa língua. A mistura de línguas de fato melhora o desempenho, mas de forma lenta e comprometendo a confiança da rede nas próprias predições a partir de certo volume de dados. O melhor caminho caso seja necessário misturar dados para tentar melhorar a efetividade é usar o mesmo número de dados em todas as línguas que serão necessárias. Todos esses resultados foram obtidos usando como *encoder* uma rede treinada com diversas línguas com um alto volume de dados, usar uma rede sem isso pode levar a resultados diferentes.

6 Conclusão

Sistemas de questão e respostas automáticas para plataformas de *e-commerce* podem se beneficiar do reuso de conjunto de dados de uma língua para responder questões em outra língua. Contudo, a literatura ainda é muito escassa nesses estudos. Este trabalho propôs um sistema que usa a similaridade entre questões em linguagem natural para responder automaticamente questões em plataformas de comercio eletrônico. Investigamos diversos experimentos sobre o uso de conjunto de dados em diferentes línguas para o treinamento de uma rede sobre similaridade entre questões. Concluímos que a mistura de línguas num conjunto de dados, para a tarefa em questão, não atrapalha a efetividade da rede, podendo inclusive melhora-lo em alguns casos. Trabalhos futuros envolvem novos experimentos para averiguarmos outros fatores que podem auxiliar no uso de conjunto de dados multilínguas para o treinamento de redes de similaridade.

Agradecimentos

GoBots LTDA.

Referências

[1] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey, 2019.

[2] H. Bast, Björn Buchhold, and Elmar Haussmann. Semantic search on text and knowledge bases. *Found. Trends Inf. Retr.*, 10:119–271, 2016.

[3] Nihar Bendre, H. Terashima-Marín, and Peyman Najafirad. Learning from few samples: A survey. *ArXiv*, abs/2007.15484, 2020.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[5] Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, C. Tar, Yun-Hsuan Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *ArXiv*, abs/1803.11175, 2018.

[6] P. Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers: 2nd edition (with python examples). *ArXiv*, abs/2004.04523, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Anna Koroleva, Sanjay Kamath, and Patrick Paroubek. Measuring semantic similarity of clinical trial outcomes using deep pre-trained language representations. *Journal of Biomedical Informatics: X*, 4:100058, 2019.

[9] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[11] Mridul Mishra and Jaydeep Viradiya. Survey of sentence embedding methods, 04 2019.

[12] Eetu Mäkelä. Survey of semantic search research. 07 2008.

[13] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Keras Tuner. `https://github.com/keras-team/keras-tuner`, 2019.

[14] Nouha Othman, Rim Faiz, and Kamel Smaïli. Improving the community question retrieval performance using attention-based siamese lstm. In Elisabeth Métais, Farid Meziane, Helmut Horacek, and Philipp Cimiano, editors, *Natural Language Processing and Information Systems*, pages 252–263, Cham, 2020. Springer International Publishing.

[15] C. S. Perone, Roberto Silveira, and Thomas S. Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *ArXiv*, abs/1806.06259, 2018.

[16] Tharindu Ranasinghe, Constantin Orasan, and R. Mitkov. Semantic textual similarity with siamese neural networks. In *RANLP*, 2019.

[17] H. Edwin Romeijn. *Random search methodsRandom Search Methods*, pages 3245–3251. Springer US, Boston, MA, 2009.

[18] Lakshay Sharma, L. Graesser, Nikita Nangia, and Utku Evci. Natural language understanding with the quora question pairs dataset. *ArXiv*, abs/1907.01041, 2019.

[19] Vaishali Singh and Sanjay K. Dwivedi. Question answering: A survey of research, techniques and issues. *Int. J. Inf. Retr. Res.*, 4(3):14–33, July 2014.

[20] Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. BERT-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75, Barcelona, Spain, December 2020. Association for Computational Linguistics.

[21] H. Zhang, T. Wang, Xiaonan Meng, and Yi Hu. Improving semantic matching via multi-task learning in e-commerce. In *eCOM@SIGIR*, 2019.