USO DE ADAPTAÇÃO DE DOMÍNIO E INFORMAÇÃO CONTEXTUAL EM SISTEMAS DE PERGUNTA–RESPOSTA

GIANLUCCA LODRON ZUIN

USO DE ADAPTAÇÃO DE DOMÍNIO E INFORMAÇÃO CONTEXTUAL EM SISTEMAS DE PERGUNTA–RESPOSTA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Adriano Veloso
Coorientador: Luiz Chaimowicz

Belo Horizonte
Novembro de 2017

Zuin, Gianlucca Lodron

Z94u Uso de Adaptação de Domínio e Informação Contextual em Sistemas de Pergunta−Resposta / Gianlucca Lodron Zuin. — Belo Horizonte, 2017
xvi, 71 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais
Orientador: Adriano Veloso
Coorientador: Luiz Chaimowicz

1. Computação — Teses. 2. Aprendizado de Máquina — Teses. 3. Redes Neurais (Computação).
4. Pergunta-Resposta. 5. Adaptação de Domínio.
I. Orientador. II. Coorientador. III. Título.

CDU 519.6*82(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Uso de adaptação de domínio e informação contextual em sistemas de pergunta-resposta

GIANLUCCA LODRON ZUIN

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ADRIANO ALONSO VELOSO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. LUIZ CHAIMOWICZ - Coorientador
Departamento de Ciência da Computação - UFMG

PROFA. AGMA JUCI MACHADO TRAINA
Departamento de Ciências de Computação e Estatística - USP

PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 14 de Novembro de 2017.

Este trabalho é dedicado a todos que me apoiaram e ajudaram durante essa jornada. Eles me forneceram as condições vitais para a confecção desse trabalho e acredito que sem eles nada disso teria sido possível. A todos vocês, sou extremamente grato.

Agradecimentos

Aos meus professores que me guiaram nesta jornada, por seus ensinamentos valiosos e as oportunidades promissoras. Em especial, agradeço ao professor Luiz Chaimowicz, por ter me acompanhado e orientado desde a graduação, tendo me auxiliado no início de minha carreira acadêmia durante meu primeiro trabalho científico em 2015 e ao professor Adriano Veloso, por ter aceitado entrar na jornada que foi este trabalho após seu começo, mostrando sempre as armadilhas associadas ao problema proposto e fornecendo soluções.

Aos meus pais, Romanelli e Elenice, não somente por terem fornecido condições que me permitiram focar no Mestrado, mas também por serem sempre grandes exemplos. Agradeço também por toda a ajuda que forneceram durante a escrita deste trabalho, permitindo que eu aproveitasse de sua vasta experiência. Da minha mãe, herdei a aproximação com a Matemática, do meu pai e do meu tio Ronaro, o trilhar pelos caminhos da Ciência da Computação.

À minha avó Leila, pelo carinho, apoio e incentivo em todos os momentos da minha vida acadêmica.

À minha namorada, Laura Cristina, pela compreensão nos momentos que tive de me concentrar na pesquisa e pelo companheirismo de maneira geral. Por ser minha principal válvula de escape do *stress* do dia-a-dia e por estar presente nos momentos mais importantes de minha vida.

Aos meus amigos, por sempre me forneceram bons sorrisos e momentos de alegria. Suas pequenas distrações muitas vezes permitiram que eu observasse o problema tratado de outro ângulo e chegasse a novas soluções.

Finalmente, à CAPES e a todos os funcionários do PPGCC. A CAPES forneceu o apoio financeiro que me permitiu dedicar exclusivamente ao programa de mestrado.

Resumo

Um sistema de pergunta−resposta (*question answering*, QA) é um sistema capaz de receber como entrada uma quantidade não restrita de questões em linguagem natural e que fornece uma resposta. Geralmente coletamos dados de diversas fontes para montarmos um Corpus adequado para o aprendizado de modelos multi-domínio de pergunta−resposta. Este tipo de sistema requer que o modelo seja capaz de realizar compreensão de linguagem natural, o que implica na necessidade de grandes bases de dados. Uma maneira simples de aliviar a demanda de dados é restringir o domínio abordado pelo QA, levando assim à modelos específicos. Embora o aprendizado de modelos de QA em um único domínio ainda seja uma tarefa desafiadora devido à escassez de dados de treinamento suficientes no tema de interesse, podemos obter instâncias adicionais por meio de domínios relacionados. Este trabalho investiga abordagens de adaptação a fim de obter vários modelos especializados em cada domínio alternativamente a aprender um modelo único de amplo domínio. Demonstra-se ainda que isso pode ser alcançado estratificando-se uma base original, sem a necessidade de buscar dados adicionais ao contrário de outras abordagens da literatura. Este trabalho propõe uma rede neural que explora o uso conjunto de redes convolucionais e recorrentes. Características gerais dos temas são compartilhadas enquanto características específicas dos domínios são aprendidas. Isso permite realizar a adaptação dos modelos utilizando diversos tipos de domínio fonte. São consideradas diferentes abordagens de transferência e de divisão de domínios desenvolvidas para aprender modelos de QA tanto em nível de *spans*, quanto em nível de sentenças. Observou-se que a adaptação ao domínio resulta em ganhos de desempenho, em especial ao nível de sentenças. Observou-se também que podemos ter um aumento considerável no desempenho do modelo baseado em *spans* ao utilizar a informação de contexto presente no QA de sentenças.

Palavras-chave: Pergunta-Resposta, Redes Neurais Profundas, Adaptação de Domínio, Transferência de Aprendizado, Integração de Contexto.

Abstract

A question answering (QA) system is a system that receives a question in natural language as input and that attempts to provide an answer. Corpora used to learn open-domain QA models are typically collected from a wide variety of topics or domains. Since QA requires understanding natural language, open-domain QA models generally need very large training corpora. A simple way to alleviate data demand is to restrict the domain covered by the QA model, leading thus to domain-specific QA models. While learning improved QA models for a specific domain is still challenging due to the lack of sufficient training data in the topic of interest, additional training data can be obtained from related topic domains. Thus, instead of learning a single open-domain QA model, this work investigates domain adaptation approaches in order to create multiple improved domain-specific QA models. It is also shown that this can be achieved by stratifying the source dataset, without the need of searching for complementary data, unlike many other domain adaptation approaches. This work proposes a deep architecture that jointly exploits convolutional and recurrent networks for learning domain-specific features while transferring domain-shared features. That is, transferable features to enable model adaptation from multiple source domains. It is considered different transference and domain selection approaches designed to learn span-level and sentence-level QA models. The findings show that domain-adaptation improves performance, specially in sentence-level QA. It is also shown that span-level QA benefits from contextual information present in the sentence models.

Keywords: Question Answering, Deep Neural Networks, Domain Adaptation, Transfer Learning, Context Integration.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Um sistema de pergunta−resposta (*question answering*) é um sistema capaz de receber como entrada uma quantidade não restrita de questões em linguagem natural e que tenta fornecer uma resposta ao buscar em dados armazenados [Hirschman & Gaizauskas, 2001]. Apesar da grande quantidade de trabalhos na área desde os anos 2000, pesquisas sobre esse tema são bem mais antigas. Alguns dos primeiros e mais famosos sistemas de pergunta−resposta foram o BASEBALL [Green Jr et al., 1961], que respondia às perguntas sobre a liga de baseball americana e LUNAR [Woods & Kaplan, 1977], treinado com dados sobre rochas e solo lunares coletados pela Apollo 11 e com 90% de acurácia em suas respostas. Vários outros sistemas já existiam nessa época, como os relatados no trabalho de Simmons [Simmons, 1965], onde 15 programas capazes de responder perguntas através de texto são analisados. Nos 55 anos desde o desenvolvimento do BASEBALL muito mudou, não só na área de pergunta−resposta, mas em toda a grande área de processamento de linguagem natural.

Considera-se, portanto, a tarefa de aprender modelos de respostas à perguntas de amplo domínio (doravante modelos de pergunta−resposta ou QA, *Question Answering*). Isto é, modelos que encontram respostas às perguntas formuladas sobre um tópico qualquer em coleções não estruturadas de documentos [Bordes et al., 2015a]. É trabalhoso montar corpus grandes o suficiente para permitir o aprendizado de modelos de QA multi-domínio, pois essas bases devem abranger uma grande variedade de assuntos. Alternativamente, restringindo-se o domínio da pergunta, a demanda de dados se torna significativamente menor [Ghung et al., 2004]. A idéia principal explorada neste trabalho é a de que um modelo multi-domínio pode ser decomposto em vários modelos menores específicos. Podemos aprender cada um deles de forma independente e, posteriormente, os unir gerando assim um modelo multi-domínio aprimorado.

Assumiu-se que perguntas podem ser mapeadas para domínios, estes associados a sua temática implícita específica [Bhatia et al., 2016]. Pode-se, portanto, direcionar cada instância das bases de treino e avaliação para um modelo que abrange seu tópico. Nesse caso, podemos considerar que um domínio é definido em função tanto de palavras comuns compartilhadas por todos os domínios quanto de palavras que são específicas para domínios individuais [Chen & Zhang, 2013]. Esse conjunto de palavras específicas gera uma temática comum entre as perguntas e parágrafos relacionados. Tendo definido os domínios de cada tópico, os modelos de QA são finalmente aprendidos. Aprender um modelo especializado em um domínio ainda é uma tarefa desafiadora em vista da escassez de dados sobre cada tema de interesse. A abordagem proposta utiliza dados de treinamento adicionais derivados de domínios relacionados.

Este trabalho explora a adaptação do domínio com o intuito de melhorar o sistema de QA no geral. Foi empregada uma arquitetura profunda [Tan et al., 2016] composta de redes neurais convolucionais (CNN) e redes recorrentes bidirecionais, mais especificamente uma biLSTM. A principal hipótese explorada é que a combinação dessas estruturas oferece uma perspectiva semântica complementar do texto, explorando tanto aspectos espaciais quanto temporais dos pares de perguntas e respostas. Enquanto a estrutura da CNN extrai características espaciais em diferentes níveis de abstração, a LSTM é capaz de focar nas informações contextuais, modelando as dependências textuais e permitindo uma análise de padrões anteriores e posteriores de cada segmento das perguntas e respostas [Seo et al., 2016]. Discute-se diferentes abordagens de transferência de aprendizado, alternando a escolha de quais camadas congelamos ou atualizamos.

Embora o objetivo principal seja o QA em nível de *spans*[1], também considerou-se o problema de QA ao nível de sentenças, onde o objetivo é retornar a frase que contém a resposta correta. Esse problema é ligeiramente mais fácil e os modelos que abordam essa tarefa têm um desempenho superior. Tendo em vista esta característica aliada ao fato de que muitas vezes os modelos de QA apresentam um F1[2] alto, mesmo com uma acurácia ligeiramente mais baixa, também propõe-se condicionar a escolha dos *spans* candidatos usando a saída do modelo em nível de sentença. Ou seja, escolhe-se respostas relevantes que também estejam relacionadas a passagens relevantes. A seleção de respostas foi formulada como um problema de busca envolvendo relevância de sentenças e *spans*.

[1]Pequeno segmento de texto contendo apenas algumas poucas palavras. No cenário de linguagens naturais, consiste em uma seção de uma sentença.

[2]F1 ou F-score é uma medida da acurácia de um teste. Esta métrica considera tanto a precisão quanto a revocação, calculando a média harmônica de ambas.

Realizou-se ainda um conjunto amplo de experimentos usando o repositório de dados de amplo domínio do SQuAD [Rajpurkar et al., 2016]. O SQuAD fornece um ambiente de testes desafiador para avaliar os modelos propostos. Foram definidos então os domínios de cada tópico e foram construídos os modelos de QA específicos dos domínios correspondentes. Os resultados indicam que a adaptação é efetiva, levando a ganhos de acurácia que chegam a 20% em alguns domínios. Em média, todos os modelos têm um aumento de acurácia de 10% ao realizar a adaptação do domínio. O condicionamento de sentenças também é eficaz, já que observou-se um aumento de 40% no desempenho de QAs em nível de *spans* ao realizar o condicionamento às sentenças.

1.1 Motivação

De acordo com Manning [Manning et al., 2008], sistemas de recuperação de informação (*Information Retrieval*, IR) têm como objetivo encontrar dados de natureza não estruturada que satisfaçam uma necessidade pesquisando dentro de grandes coleções como, por exemplo, seções de texto dentro de documentos. Além disso, afirma que recuperação de informação tem rapidamente se tornado a forma dominante de acesso à informação, ultrapassando a tradicional busca em bancos de dados. Porém, esse tipo de acesso possui desvantagens. No modelo IR padrão, as consultas são pequenas entradas contendo palavras chave extremamente específicas e pouco abrangentes. Em contrapartida, as saídas tendem a ser extensas e a informação presente encontra-se difundida entre listas de documentos e passagens pertinentes. Um sistema de pergunta−resposta visa remediar esses dois grandes empecilhos: as entradas podem ser abrangentes e as saídas devem ser compactas e diretas.

Vários sistemas de QA têm surgido. Entre eles podemos mencionar o *Wolfram Alpha*[3], baseado em uma coleção de dados curados e particularmente famoso pelo seu arcabouço matemático; o *EAGLi*[4], especializado no domínio de saúde; e o próprio *Google*, como ilustrado na Figura 1.1. Dentre os mais famosos, destaca-se o sistema da IBM, *Watson* [Ferrucci et al., 2010], que derrotou os dois maiores campeões do programa de televisão de perguntas e respostas *Jeopardy* [Hanna, 2011]. Em 2013, foi anunciado que seria utilizado numa aplicação comercial para a decisões no tratamento de pacientes com câncer de pulmão, [Upbin, 2013].

Aplicações de sistemas de resposta incluem suporte a usuários, confecção de agentes críveis, comunicação entre agentes, automatização de sites de resposta à perguntas (como o *Yahoo! Answers*), dentre outros. Com isso em vista, a área de

[3] `https://www.wolframalpha.com`
[4] `http://eagl.unige.ch/EAGLi`

Figura 1.1: Exemplo de consulta que utiliza um sistema de pergunta−resposta integradas em sua plataforma.

pergunta−resposta é um cenário importante de pesquisa com amplas aplicações e utilizado em diferentes projetos.

Redes neurais convolucionais e recorrentes são eficazes em tratar do problema de pergunta−resposta abordando a tarefa de classificação de respostas candidatas [Feng et al., 2015a, Tan et al., 2015, Yin et al., 2016]. Estas redes têm sido bastante utilizadas na literatura em experimentos envolvendo análise de áudio e de imagens. Nesses cenários, elas têm tido um ganho considerável quando utilizadas em conjunto com técnicas de transferência de aprendizado [Ahmed et al., 2008, Coutinho et al., 2014, Shin et al., 2016]. Adaptação de domínio similar à maneira empregada é algo inédito no problema de QA. A similaridade entre os cenários em que essa abordagem foi empregada e o sucesso das redes descritas sugerem que resultados significativos possam ser obtidos.

1.2 Definição do problema

Dada uma pergunta q e uma lista de respostas candidatas $\mathcal{A}_{\text{II}} = \{a_1, a_2, \ldots, a_n\}$, desejamos ordenar essa lista em função da relevância $f(q, a_i)$. Dado também um conjunto de domínios $\mathcal{D} = D_1, D_2, D_d$, assumimos que cada pergunta está associada a um domínio relacionado a sua temática. Assim, podemos expressar o conjunto de perguntas $\mathcal{Q}$ como $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_d\}$ e onde Q_i representa as perguntas associadas a algum dos d domínios. Logo, desejamos encontrar as funções f_d que maximizem a acurácia do sistema em cada domínio, representado pela equação 1.1. Ou seja, as funções que melhor classifiquem as respostas corretas para cada uma das perguntas de nossa base,

levando em consideração todo o conjunto de respostas incorretas.

$$f_d = argmax(f(q_i, A_d)) \quad | \quad d \in \mathcal{D}, q \in Q_d \tag{1.1}$$

1.3 Objetivos

O objetivo principal deste trabalho é implementar um modelo multi-domínio de pergunta−resposta que utilize adaptação de domínio. Os demais cenários que empregam essa abordagem apresentam ganhos consideráveis de desempenho e este trabalho visa relatar que isso também se aplica em sistemas de QA. Durante o desenvolvimento do trabalho, verificou-se que o desempenho do modelo base desenvolvido estava aquém dos *baselines* encontrados no estado-da-arte. Desta forma, determinou-se um objetivo secundário de superar esses *baselines*. Para tal, foi treinado um modelo adicional focado em informações contextuais de cada pergunta e que também utilizou adaptação de domínio.

Destacam-se os objetivos específicos:

- Realizar uma pesquisa extensiva sobre adaptação de domínio e redes neurais aplicadas a sistemas de pergunta−resposta a fim de averiguar os diferentes métodos existentes na literatura.
- Implementar um modelo de pergunta−resposta que tenha desempenho comparável aos demais métodos de estado-da-arte.
- Utilizar adaptação de domínio no modelo proposto e avaliar o ganho de desempenho.
- Explorar diferentes abordagens para a realização de adaptação de domínio. Ainda, comparar o desempenho de cada uma e suas respectivas vantagens e desvantagens.

1.4 Contribuições

A principal contribuição deste trabalho é elucidar que arquiteturas profundas envolvendo QAs podem se beneficiar da adaptação do domínio usando os temas das perguntas. De maneira geral, também destaca-se as seguintes contribuições:

- Embora o modelo proposto tenha semelhança com modelos anteriores para QA [Tan et al., 2016, Feng et al., 2015b], uma grande diferença é que propõe-se lidar

com o problema tanto em nível de sentença quanto de *spans*. Em vez de classificar a frase e posteriormente extrair a resposta, combinar a probabilidade de uma resposta específica ser a correta com a probabilidade de uma sentença específica ser a correta, dado que esta frase contém a resposta.

- Condicionar a escolha do *span* candidato usando informação de contexto presente no problema de sentenças leva a um ganho considerável de desempenho. Além disso, estratégias muito simples para esse condicionamento são efetivas.

- Propor uma estratégia de divisão e conquista para aprender um modelo aprimorado de QA multi-domínio. Basicamente, separar os dados em domínios temáticos a partir dos quais modelos de QA específicos são treinados. Depois selecionar o modelo que tenha a melhor performance para cada instância do conjunto de avaliação, emulando assim uma configuração de multi-domínio.

- Mostrar que o modelo proposto leva a melhorias substanciais no conjunto de dados do SQuAD.

- Ilustrar que não é necessário buscar dados adicionais para realizar uma abordagem efetiva de adaptação de domínio. Podemos estratificar nossos dados e dividi-los em várias tarefas.

1.5 Organização da Dissertação

Esta dissertação é dividida em 6 capítulos incluindo o atual. O Capítulo 2 contêm uma revisão detalhada das diversas técnicas e ferramentas que foram empregadas. Ele explica conceitos básicos de redes convolucionais e recorrentes assim como o algoritmo K-means e a intuição por trás de *embeedings*.

No Capítulo 3 é apresentada uma sumarização dos diversos trabalhos publicados na literatura cujos temas estão relacionados ao problema destacado. O capítulo é dividido em três partes: uma visão histórica de sistemas de QA, redes neurais aplicadas a QAs e alguns trabalhos que abordam transferência de aprendizado e adaptação de domínio.

O modelo proposto é introduzido no Capítulo 4. Também é levantada uma discussão sobre as diferentes abordagens de adaptação de domínio assim como técnicas para a divisão das bases de treino e teste. No Capítulo 5 são apresentadas sete questões de pesquisa e conduzidos experimentos a fim de responder cada uma delas.

Finalmente, o Capítulo 6 traz a conclusão e as considerações finais. Alguns dos resultados obtidos no Capítulo 5 são destacados e discutidos em maior profundidade. A dissertação é finalizada com as propostas para o futuro.

Capítulo 2

Referencial teórico

Neste capítulo são introduzidos os vários conceitos e métodos empregados nesse trabalho. O objetivo é sumarizar o conhecimento necessário para compreender as técnicas utilizadas. Discute-se, principalmente, a arquitetura geral e as intuições por trás de uma rede neural tradicional e, então, são detalhados os dois tipos utilizados: redes convolucionais e redes recorrentes. Também é detalhado o funcionamento de outras técnicas utilizadas mas não necessariamente relacionadas com redes neurais: *embeddings*, divergência de Kullback–Leibler e o método de clusterização K−means.

2.1 Redes Neurais

O cérebro humano é composto por bilhões de neurônios, cada um deles sendo constituido principalmente por dentritos e axônios. Quando um neurônio é ativado em resposta a algum estímulo, um impulso elétrico é gerado e propagado pelo axônio até suas extremidades, que estão conectadas a múltiplos dentritos de outros neurônios. Essa conexão é denominada sinapse. Os neurônios vizinhos, por sua vez, avaliam a combinação dos sinais sendo recebidos em função de um certo nível de ativação, determinando se também serão ativados. Eles então progagam essa informação pelos seus axônios e o processo é repetido sucessivamente pelo sistema nervoso [Harvey, 1994]. Dessa forma, as várias sinapses formam uma rede e a geração e propagação de impulsos elétricos por ela resumem o funcionamento do nosso sistema nervoso.

Uma rede neural artificial (ANN, *Artificial Neural Networs*) é *'um sistema composto por um número de elementos de processamento simples, altamente interligados, que processam a informação pela sua resposta de estado dinâmica a entradas externas.'*[Caudill, 1989]. Ela é um método bioinspirado, constituído de um conjunto de funções não-lineares simples, interligadas entre si, que simulam a relação entre neurô-

nios e axônios no cérebro. A Figura 2.1 ilustra a comparação entre um neurônio artifical de uma rede neural e neurônios reais. O tipo mais básico de rede neural artificial são aquelas que utilizam um classificador binário como ativação de cada neurônio. Ou seja, se o valor de entrada é maior ou igual ao esperado a saída do neurônio é 1, caso contrário, a saída propagada é 0. Esse algoritmo foi desenvolvido em 1957 e foi denominado *perceptron* [Rosenblatt, 1957]. Apesar de arquiteturas que envolvem um único *perceptron* serem incapazes de resolver problemas que não sejam linearmente separáveis, redes com múltiplas camadas de *perceptrons* (MLP, *Multi-layer Perceptron*) não possuem essa limitação e impulsionaram o interesse de pesquisadores em redes neurais.

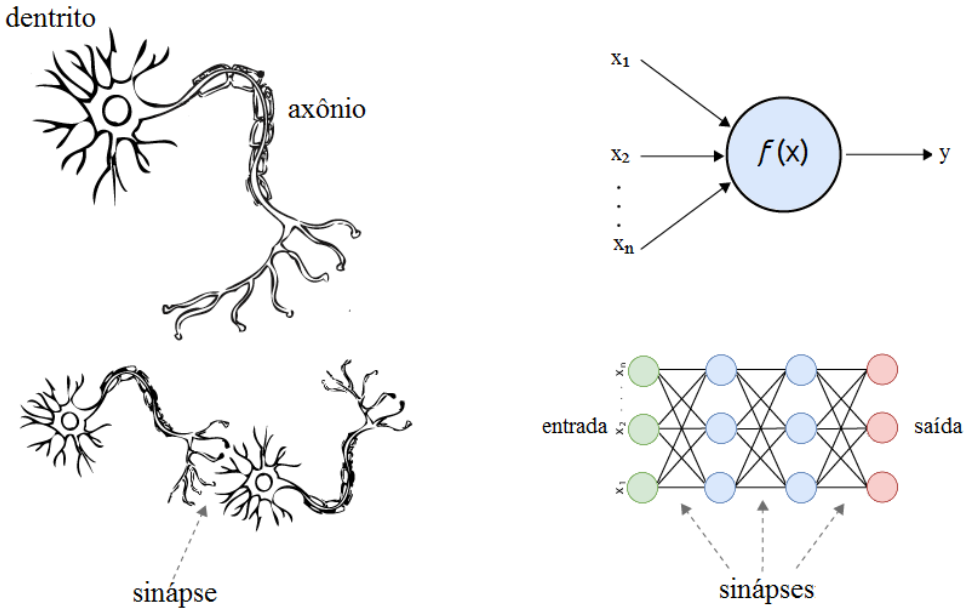


Figura 2.1: Imagem comparativa um neurônio real e um artificial, assim como entre uma rede neural artificial e uma sinapse contento dois neurônios. Ilustração de um neurônio real retirada do livro *Brain Power: Grades 6-9* [NIDA, 2007].

Em uma rede tradicional, temos principalmente três tipos de camadas. A primeira delas, denominada *'camada de entrada'*, recebe os dados a serem interpretados pela rede. A segunda camada, a *'camada escondida'*, recebe as saídas da primeira camada e tem como objetivo processá-los e gerar uma abstração. A existência de múltiplas camadas escondidas aumenta o poder de abstração da rede, mas a deixa mais complexa e propícia a sofrer *overfitness*. Isto é, ela fica mais propícia a se especializar demais nos dados de teste e perder parte de sua capacidade de generalização em dados ainda não vistos. A última das camadas combina a saída dos neurônios da camada escondida mais profunda e busca compilar os dados em uma representação qualquer. A Figura 2.2 ilustra a arquitetura de uma rede neural simples. Cada aresta da rede possui um peso associado e a etapa de treino consiste em otimizá-los a fim tornar a saída da rede

o mais próxima possível da esperada. Isso é feito por meio do algoritmo de descida de gradiente, uma função da derivada dos pesos da rede e o erro entre as saídas esperadas e obtidas, que quantifica a atualização dos pesos. A métrica que avalia a qualidade da rede treinada é encontrada de acordo com alguma função de otimização utilizada.

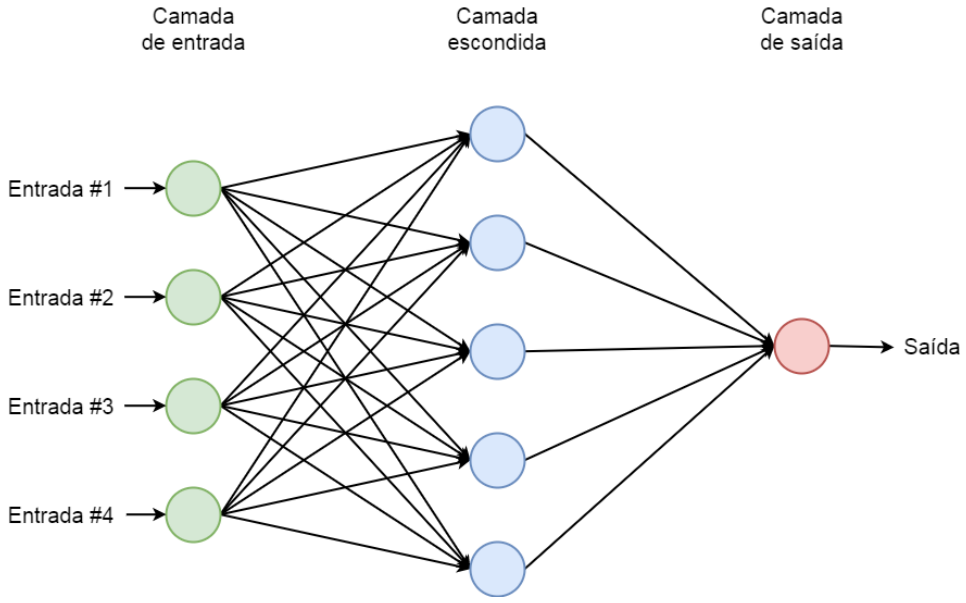


Figura 2.2: Exemplo de uma rede neural *feedforward* simples com uma camada escondida e um único neurônio de saída.

2.2 Redes Convolucionais

As redes neurais convolucionais (CNN) são variantes das primeiras redes contendo múltiplos *perceptrons* e foram inspiradas nos primeiros trabalhos de Hubel e Wiesel no córtex visual de gatos e macacos [Hubel & Wiesel, 1968]. O trabalho mostra que existe um arranjo complexo de neurônios que são sensíveis às pequenas sub-regiões do campo visual e que respondem individualmente a diferentes padrões, como ilustrado pela Figura 2.3. Em particular, identifica-se que existem dois tipos de células. A primeira delas, as células simples, são especializadas em identificar linhas retas possuindo determinadas orientações. Uma característica interessante é que células vizinhas analisam sub-regiões do campo visual adjacentes e possuem uma pequena área de sobreposição. O segundo tipo de célula, células complexas, possui um campo receptivo maior e não é sensível às regiões. Ambas as células atuam como filtros locais sobre o espaço de entrada e são adequadas para explorar a forte correlação espacial presente em imagens naturais. A teoria por trás das redes neurais convolucionais baseia-se fortemente nelas.

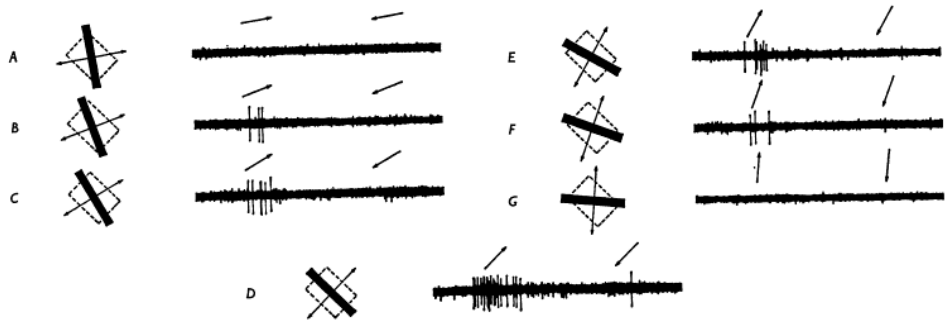


Figura 2.3: Célula complexa presente nos gatos e especializada em identificar linhas retas em movimentação numa orientação de 45°. Quanto mais próximo dessa angulatura ótima, maior a ativação. Note que a movimentação em um sentido causa um estímulo maior que no outro. A união desta célula com os demais neurônios especializados em diferentes padrões e orientações perimitem que o gato interprete o que ele vê. Imagem retirada do trabalho de Hubel [Hubel & Wiesel, 1968].

Um dos trabalhos pioneiros utilizando CNNs foi o de LeCun [LeCun et al., 1998], no qual é proposta a *LeNet* (Figura 2.4), uma rede neural profunda representada por duas camadas convolucionais intercaladas com uma camada de *pooling* e seguidas de camadas totalmente conectadas. Ela é avaliada no problema de reconhecimento de dígitos, no qual imagens de caracteres são transformadas em uma matriz onde cada célula está associada à intensidade de um pixel da fonte. Seus resultados foram extremamente promissores, mostrando que as redes convolucionais eram superiores aos demais métodos empregados até o momento. Seu trabalho impulsionou não somente a pesquisa relacionada à reconhecimento de imagens mas também aprendizado profundo.

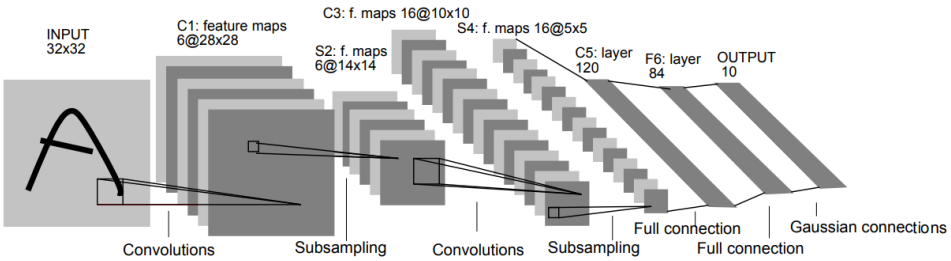


Figura 2.4: Arquitetura da LeNet5 utilizada para reconhecimento de dígitos. Ela segue todos os detalhes necessários para o funcionamento adequado de redes convolucionais explicados aseguir. Imagem retirada do trabalho de LeCun. [LeCun et al., 1998]

Podemos visualizar uma CNN como uma rede que agrupa um conjunto de convoluções, na qual cada uma delas analisa segmentos da entrada e realiza uma operação sobre esses segmentos, gerando uma saída. Os filtros de convolução são responsáveis por definir o tamanho do segmento analisado e a operação realizada sobre eles. A

Figura 2.5 ilustra um exemplo do operador de convolução utilizando um *stride* de 1x1. O *stride* controla como o filtro analisa toda a entrada. Um *stride* de NxM implica que caminhamos N unidades para identificarmos o próximo segmento no eixo horizontal e M unidades no eixo vertical. A combinação de diferentes filtros caracteriza uma CNN. Além do operador de convolução algumas das características importantes das CNNs que as permitem funcionar são a conectividade espacial e pesos compartilhados.

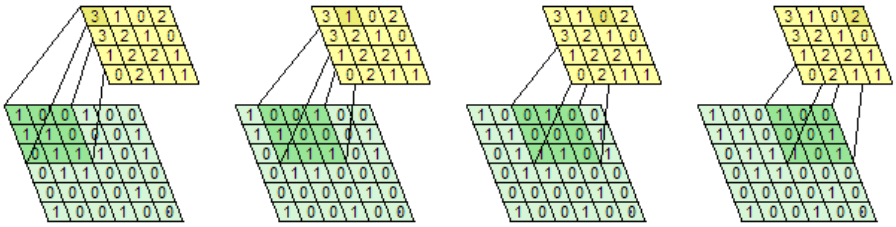


Figura 2.5: Exemplo de convolução buscando um padrão diagonal com um filtro 3x3 e com stride de 1x1.

As redes convolucionais aproveitam da correlação local dos dados forçando a conexão entre neurônios de camadas adjacentes. A Figura 2.6 ilustra essa relação. Cada neurônio das camadas convolucionais está conectado a um grupo de neurônios da camada posterior. A quantidade exata é definida em função do campo receptivo da CNN (nesse exemplo, ele tem comprimento 3) e implica no tamanho do filtro. Observa-se que o neurônio da camada mais profunda tem um campo receptivo maior em relação à entrada, ilustrado pela àrea delimitada do neurônio C.

Cada neurônio é sensível apenas a valores dentro do seu campo receptivo. A resposta do neurônio A é invariável em relação aos neurônios do campo de B. A ativação de A em contraste a B nos garante que encontramos um padrão no início do vetor de entrada. A informação é então propagada no restante da rede para as camadas superiores. Essa arquitetura garante que os filtros aprendidos tenham uma resposta mais forte para padrões locais na entrada.

Nas redes convolucionais, a profundidade da rede não apenas implica no aumento da capacidade abstração, mas também no tamanho do segmento da entrada analisado por um único neurônio. Neurônios em camadas mais rasas estão relacionados a padrões extremamente específicos mas simples. Os presentes em camadas mais profundas são responsáveis por aprender características globais dos dados, reunindo o conjunto de padrões simples, mas específicos para gerar um padrão mais complexo. Em CNNs aplicadas ao processamento de imagens, as primeiras camadas podem ser responsáveis por identificar pequenos segmentos de linhas retas em diferentes ângulos por exem-

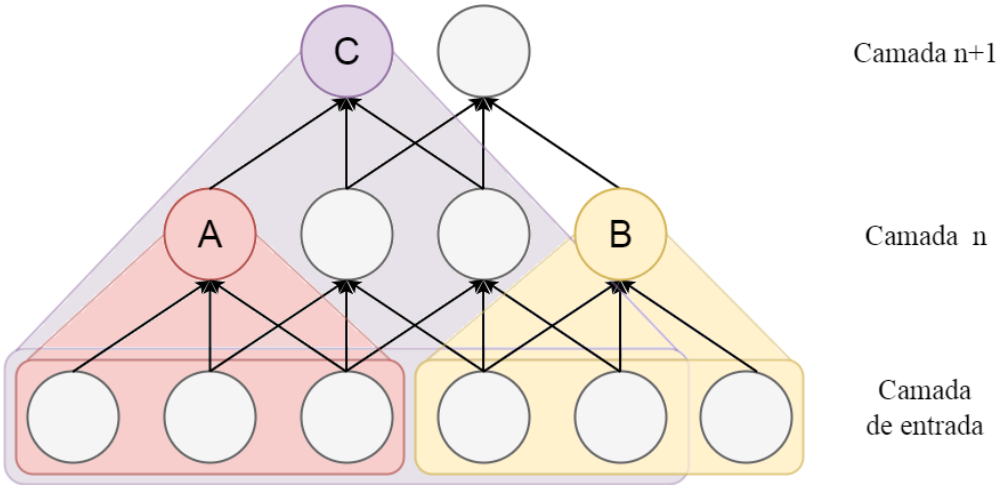


Figura 2.6: Conectividade de neurônios em diferentes camadas convolucionais. Cada neurônio é sensível apenas à mudanças dentro de seu campo receptivo. Camadas mais profundas têm um campo receptivo 'total' maior por serem sensíveis às entradas de todos os neurônios de seu campo receptivo real, como observado pelo campo do neurônio C em comparação com o de A e B.

plo, enquanto as últimas já identificam formas muito mais complexas como rostos ou objetos.

A premissa do operador de convolução é que um mesmo filtro é avaliado por toda a entrada. Usando apenas as características descritas até agora não temos essa garantia. De fato, ao realizarmos a propagação do erro durante a etapa de aprendizado, cada aresta da rede teria seus pesos atualizados individualmente. Logo, segmentos da entrada seriam avaliados de maneira diferente dos demais, derrotando o propósito das redes convolucionais. Para garantir o funcionamento adequado das convoluções, algumas características extras devem ser aplicadas. Primeiramente, cada filtro avaliado é representado por um neurônio da rede. Estes são replicados por todo seu campo de entrada, compartilhando uma mesma parametrização. O conjunto destes neurônios formam um mapa de *features*, como ilustrado pela Figura 2.7. O mapa de *features* é, portanto, a saída de um único filtro quando aplicado à camada anterior. Por conta desta característica, raramente nos referimos ao número de neurônios ao descrevermos uma rede convolucional. É mais adequado utilizar o número de filtros presentes e seu tamanho de campo receptivo.

Replicar os neurônios desta forma permite que padrões sejam detectados independentemente da sua posição no campo visual. Além disso, o compartilhamento de pesos aumenta a eficiência de aprendizagem, reduzindo consideravelmente o número de parâmetros a serem aprendidos. Essas restrições permitem que CNNs consigam uma melhor capacidade de generalização em problemas envolvendo imagens e textos,

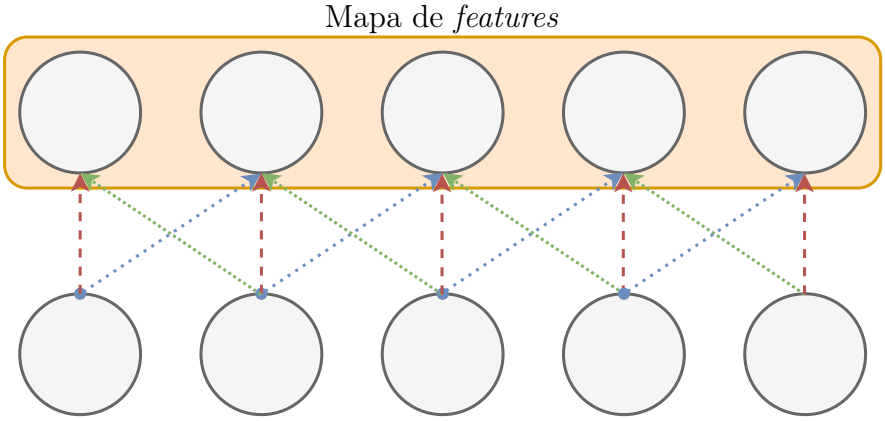


Figura 2.7: Compartilhamento de pesos em redes convolucionais. Arestas com a mesma cor e padrão compartilham os mesmos pesos e viéses. O conjunto de neurônios especializados em identificar uma certa coleção de padrões constituem o mapa de *features* e representam a saída de um filtro da camada convolucional.

exagerando nas correlações locais que dados em problemas dessa natureza geralmente possuem.

2.3 Long Short-Term Memory

Redes de *Long Short-Term Memory* (LSTM) são uma extensão de redes recorrentes (RNN) que tinham como objetivo remediar o problema da dissipação dos gradientes [Hochreiter & Schmidhuber, 1997]. Diferente de outras redes neurais, a decisão de uma rede recorrente atingida numa iteração $t-1$ afeta a decisão no momento t. Essas redes recebem duas entradas: o presente (este sendo o exemplo avaliado) e o passado recente. A sua combinação produz a resposta para novos dados.

A intuição por trás desse tipo de rede é que os seres humanos não ignoram o passado, eles não começam todo um novo processo de raciocínio a cada instante. Ao interpretar um evento, levamos em consideração várias situações anteriores pelas quais passamos. Ao tentar intepretar uma frase, as palavras lidas até o momento influenciam na nossa compreensão. A informação persiste ao longo do tempo.

Redes neurais tradicionais não têm a capacidade de realizar esse tipo de abstração, mas redes recorrentes abordam essa questão. Em sua arquitetura elas possuem laços como ilustrado na Figura 2.8, permitindo que a informação persista. Uma fração da rede examina o segmento da entrada referente ao instante t e retorna uma saída. O laço presente permite que as informações concluídas sejam passadas adiante no tempo, permitindo que a tomada de decisão das entradas do instante $t+1$ levem em consideração a saída do instante t.

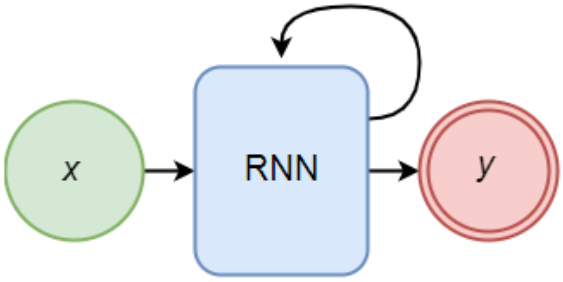


Figura 2.8: Modelagem básica de uma RNN: células possuem laços que propagam a informação ao longo do tempo.

Uma modelagem baseada em uma RNN frequentemente se depara com o problema do desaparecimento do gradiente. Menos informação sobre o passado distante é propagada a cada iteração do laço da RNN. No cenário de análise de textos, relações entre palavras muito distantes nas frases podem acabar se dissipando ao longo da camada. Esse problema tende a ser remediado pela principal característica de uma LSTM: ela guarda informação além do passado recente, a partir de uma célula de memória. Dados podem ser tanto armazenados nessa célula como também sobreescritos, lidos ou esquecidos por completo. A idéia principal por trás de LSTMs é que cada uma de suas células possui um estado, este podendo ser alterado utilizando parte dos dados armazenados em sua memória referentes a exemplos analisados no passado. A quantidade de dados utilizada é controlada por "portas". Outras portas também controlam quanto dos dados da memória devem ser atualizados. A figura 2.9 ilustra a arqutetura de uma célula da LSTM.

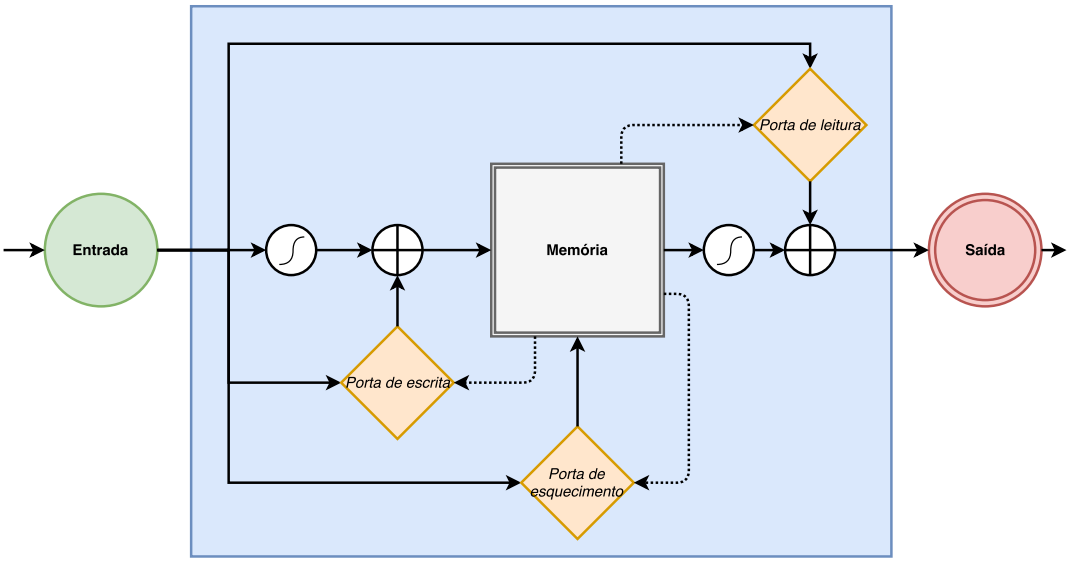


Figura 2.9: Arqutetura de uma célula da LSTM. As portas controlam quanto da memória (linhas pontilhadas) é passado adiante.

A porta de escrita destacada na Figura 2.10 tem como objetivo decidir que parcela da informação da entrada será guardada na memória da célula. Podemos dividir o processo da porta de escrita em duas etapas. Primeiramente, decidimos quais valores serão atualizados. A concatenação do estado anterior da célula e o vetor de entrada atual serão alimentados a uma função não-linear. O resultado dessa operação nos retorna um vetor onde cada elemento possui um valor real. Utilizamos esses valores como a probabilidade de gravar cada elemento do vetor de entrada na memória, atualizando-a, portanto, de maneira estocástica.

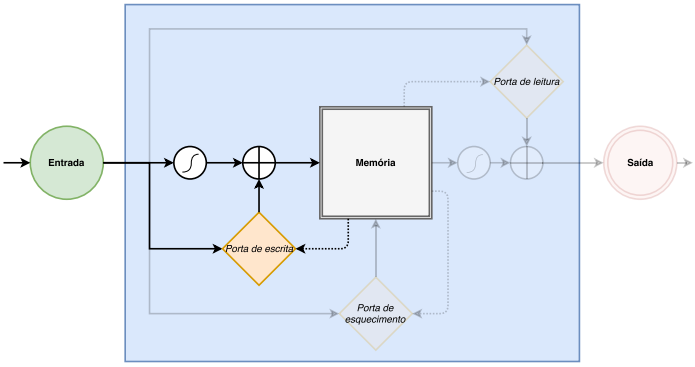


Figura 2.10: Elementos da LSTM relacionados à porta de escrita.

A segunda etapa foca em como atualizar o estado atual da célula. Processamos a mesma entrada usada na etapa anterior, mas utilizando outra função não-linear e conjunto de pesos. A idéia é que geremos a contribuição de cada elemento da entrada para o estado da célula atual. Unindo as duas etapas enunciadas, temos a descrição de quais informações queremos adicionar à representação do modelo.

A porta de esquecimento, destacada na Figura 2.11, funciona de maneira similar à de escrita. A partir da concatenação do estado anterior da célula e a entrada atual, alimentados numa função não-linear, temos um vetor de probabilidades associado a cada elemento. Decidimos, então, quais unidades manteremos na memória e quais serão esquecidas.

Com a saída das portas de escrita e esquecimento, temos as informações necessárias para atualizar a memória da célula. Podemos denominar os vetores com a informação de quais valores escrever e esquecer de i e f, respectivamente, o estado da célula de C e a segunda parte da porta de escrita, responsável por quantificar a contribuição de cada entrada para a célula, como $\widetilde{C}$. Podemos, assim, encontrar o estado atual da célula C_t por:

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \tag{2.1}$$

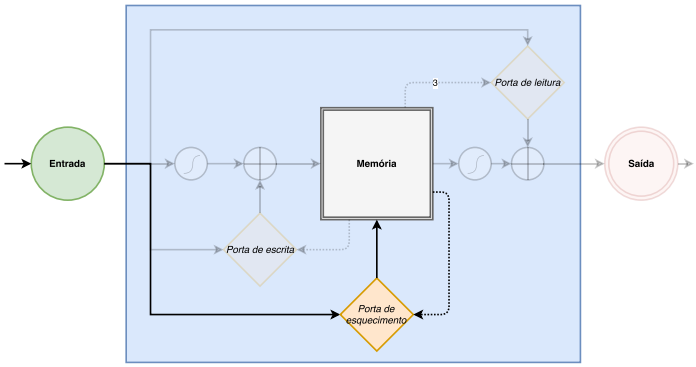


Figura 2.11: Elementos da LSTM relacionados à porta de esquecimento.

onde $\circ$ representa o produto de Hadamard[1]. Desta forma, combinamos o que queremos adicionar ao estado da célula com o que queremos retirar. O resultado é o novo estado.

Finalmente, precisamos decidir a saída da célula. Essa tarefa é realizada pela última das portas destacada na Figura 2.12. Usamos uma lógica similar à das demais portas para processar o vetor de entrada e o estado anterior, entretanto também levamos em consideração o estado atual da célula após ser atualizada. Apenas as unidades que forem ativadas tanto pelo porta de leitura quanto na memória da célula farão parte da saída final. Isso é realizado por meio das equações 2.2 e 2.3

$$o_t = \delta(W_o[x_t, h_{t-1}]) \tag{2.2}$$

$$h_t = o_t \circ tanh(C_t) \tag{2.3}$$

onde x, t e C têm a mesma nomeclatura da equação anterior e h e o representam o estado da memória e o vetor contendo as unidades ativadas, respectivamente. W_o está associado ao vetor de pesos, enquanto δ é uma função não-linear.

A visualização teórica por trás da célula de uma LSTM é diferente de sua implementação prática. Numa rede neural com camadas de LSTM, cada estado da célula é um neurônio. Assim, a "memória" da célula LSTM é emulada por uma conexão com o neurônio anterior, como ilustrado pela Figura 2.13. Adicionalmente, dependendo da arquitetura empregada, cada uma das portas é representada como um ou mais neurônios adicionais dedicados a controlar o fluxo de informação.

Uma variante importante são as redes *bidirectional Long-Short Term Memory* (biLSTM). Nelas não temos uma conexão apenas com o neurônio antecessor (emulando o passado recente), mas também com o neurônio sucessor (emulando o futuro

[1]Operador que realiza a multiplicação dos elementos de dois vetores um-a-um. Para os vetores $[a, b, c]$ e $[d, e, f]$, seu produto de Hadamard resultaria no vetor $[a * d, b * e, c * f]$

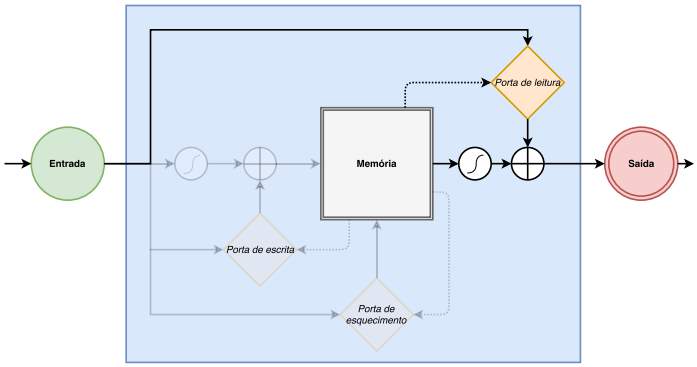


Figura 2.12: Elementos da LSTM relacionados à porta de leitura.

breve). A informação trafega ao longo da camada em ambos os sentidos. No cenário de processamento de textos, essa característica permite que a rede consiga inferir informações do estado atual utilizando dados de palavras mais adiante na frase.

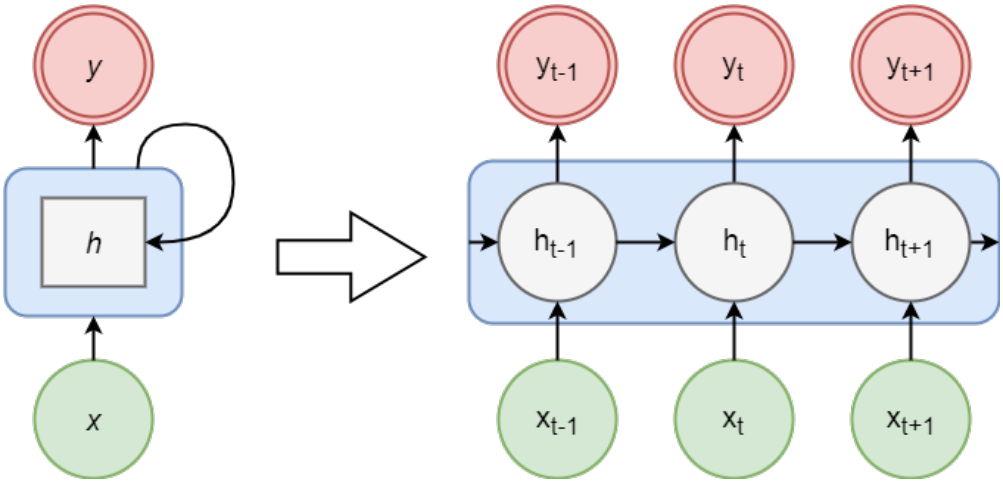


Figura 2.13: Arquitetura de uma camada de LSTM expandindo a visualização de uma célula para uma camada escondida contendo vários neurônios.

2.4 Embeddings

Uma *word embedding* W é uma função que mapeia palavras presentes em algum dicionário para vetores de alta dimensão, permitindo seu uso em uma rede neural. Uma vantagem desta técnica é que podemos inferir relações entre palavras a partir de seus vetores, algo que normalmente não seria possível se apenas atribuíssemos um índice para cada palavra. Considere, por exemplo, as palavras *lobo*, *cão* e *uiva*, presentes nas frases:

- *O lobo uiva para a lua.*

- *O cão uiva para a lua.*

Podemos esperar que os pares de palavras *lobo-uiva* e *cão-uiva* sejam próximos no espaço vetorial, visto que tem um sentido sintático comum: sons que os respectivos animais realizam. Uma forma de codificar essas relações poderia ser a partir da coocorrência de palavras. Como *lobo-uiva* e *cão-uiva* costumam aparecer juntas, elas devem ter uma relação alta e, portanto, devem estar próximas no espaço vetorial. Analogamente, uma relação similar pode ser inferida entre *uiva-lua*. Isso resulta que, indiretamente, as palavras *cão-lobo* também estão próximas vetorialmente, pois costumam aparecer em frases com a mesma estrutura e contento palavras similares. De fato, utilizar a coocorrência de palavras para quantificar os *embeddings* é o método adotado pelo *Global Vectors for Word Representation* (GloVe) [Pennington et al., 2014] e relações interessantes entre palavras surgem, como ilustrado na Figura 2.14. Nesse trabalho, utilizamos seus vetores de dimensionalidade 100 pré-treinados nos *dumps* de 2014 da Wikipedia e da quinta edição do Gigaword, que juntos constituem um vocabulário de cerca de 400.000 palavras.

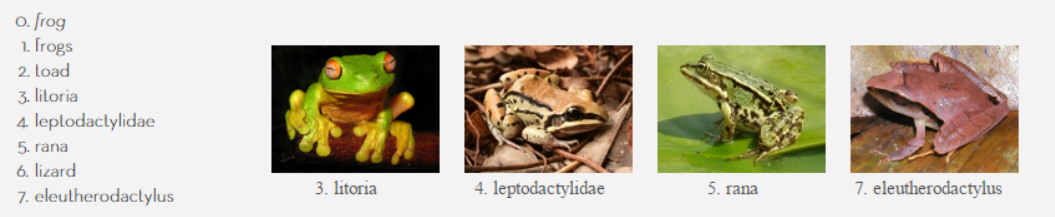


Figura 2.14: Palavras próximas de *frog* no espaço vetorial do GloVe utilizando distância de cosseno. [Pennington et al., 2014]

Podemos ir mais longe na quantificação de texto. O uso tradicional de *embeddings* é na vetorização de caracteres e palavras, mas o Doc2Vec [Le & Mikolov, 2014] propõe uma abstração maior utilizando as palavras que ocorrem em documentos. O modelo aprende uma representação simultânea tanto das palavras que compõem cada documento como os documentos em si. Esse processo é realizado principalmente por meio das sequências de palavras presentes. Nesse trabalho, utilizou-se esse método de aprendizagem de representações dos documentos para extratificar os domínios nos experimentos envolvendo K-means.

2.5 Divergência de Kullback–Leibler

A divergência de Kullback–Leibler (D_{KL}) é uma medida não simétrica da diferença entre duas distribuições de probabilidade $p(x)$ e $q(x)$ [Kullback & Leibler, 1951]. Es-

pecificamente, a $D_{KL}(p(x)||q(x))$ é uma medida da informação perdida quando $q(x)$ é usado para se aproximar $p(x)$ a partir do número esperado de bits extras necessários para codificar amostras de $p(x)$ ao utilizar $q(x)$.

Formalmente, sejam $p(x)$ e $q(x)$ duas distribuições de probabilidade de uma variável discreta X. Temos que $\sum p(x) = 1$ e $\sum q(x) = 1$. Além disso, devemos garantir que $p(x) > 0$ e $q(x) > 0$ para todo e qualquer $x \in X$. Com isso, podemos definir a divergência de Kullback-Leibler por meio da equação 2.4:

$$D_{KL}(p(x)||q(x)) = \sum_{x \in X} p(x)\, ln\frac{p(x)}{q(x)} \tag{2.4}$$

Para dois documentos de texto, podemos adotar o modelo padrão de *bag of words*, no qual um documento é representado como sendo um conjunto da contagem da ocorrência de suas palavras. De forma equivalente, um documento pode ser representado por uma distribuição de probabilidades multinomial sobre suas palavras. Para torná-la contínua, atribuímos uma probabilidade extremamente baixa às palavras que não ocorrem no documento mas fazem parte do dicionário (1^{-17}).

2.6 K-means

K−means é um método popular para análise de *clusters* e mineração de dados. Para cada observação x_i no espaço, calculamos sua distância até k centróides. Assumimos que observações próximas de um centro k_x são mais similares entre si que as próximas de um outro centro k_y, caracterizando assim uma divisão das observações em função de sua proximidade aos centros. Nesse sentido, o objetivo principal do método é particionar n observações em k *clusters* usando sua distância ao centro dessas sub-áreas do espaço [Steinhaus, 1956, MacQueen et al., 1967]. Os centros de cada *cluster* servem como observações virtuais e que sumarizam as características gerais das observações reais pertencentes a esse sub-grupo.

Formalmente, dado um conjunto de observações $(x_1, x_2, ..., x_n)$, onde cada observação representa um vetor real d-dimensional, o método de clusterização K−means busca encontrar uma partição dessas n observações em k grupos $G_1, G_2, ..., G_k$ que minimize uma métrica de instabilidade dentro desses grupos (como variância ou entropia).

O problema de encontrar um conjunto de centróides ótimo é NP−difícil. Todavia, várias heurísticas existem para solucionar o problema e são extremamente efetivas em encontrar soluções boas. Como desvantagem, essas heurísticas são propensas a cair em máximos locais e são sensíveis a escolha inicial dos centróides. Como o algoritmo tem uma convergência rápida no caso médio, uma estratégia comum é realizar várias

iterações do algoritmo modificando os centróides iniciais e então escolher o modelo de melhor desempenho (o de menor instabilidade dentro dos *clusters*). Há dois métodos comumente utilizados para a inicialização dos *clusters*: o método de Forgy [Forgy, 1965] e o de partições aleatórias.

O método Forgy escolhe aleatoriamente k observações do conjunto de dados e as usa como os centróides iniciais. O método de Partição Aleatória atribui aleatoriamente cada observação para um dos k *clusters*. Os centros são então calculados e as observações redistribuídas em função da proximidade a cada centróide. Assim, o método de Forgy tende a criar *clusters* com centros mais esparsos, enquanto o método de Partição Aleatória tende a concentrar os centróides no centro do espaço.

Uma vez definidos os centróides iniciais, o algoritmo entra na fase de refinamento iterativo, ajustando a posição de cada um dos k centróides até encontrar um ótimo local ou um determinado número de iterações ser excedido. Esse processo de refinamento é representado por uma alternância entre uma etapa de expectativa e de maximização.

Na etapa de expectativa, associamos cada observação com o *cluster* cujo centro esteja mais próximo. Na etapa de maximização, calculamos os novos centróides. Em cada iteração, certas observações transitam entre *clusters* adjacentes, resultando na modificação da posição dos centros, uma vez que eles são encontrados por meio da média das observações presentes nesse *cluster*. A Figura 2.15 ilustra uma iteração do algoritmo.

O algoritmo termina quando o sistema se estabilizar, não tendo observações transitando entre *clusters* e, portanto, não havendo atualização nos centróides. Uma característica do K−means em contraste com outros métodos de clusterização é que ele tende a criar grupos com uma quantidade semelhante de observações.

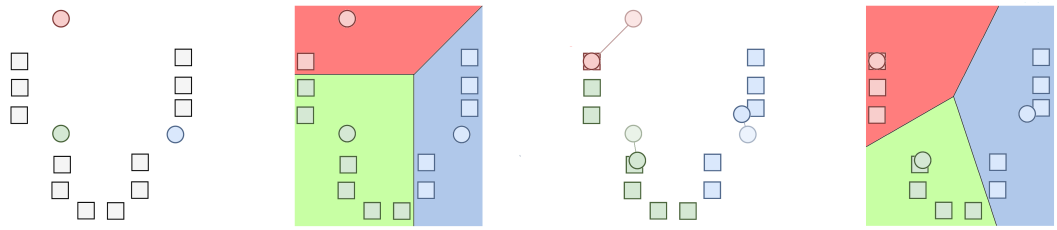


Figura 2.15: Uma iteração do algoritmo K−means. Calculados os centróides, atribuímos a cada observação um *cluster*. Pontos coloridos representam os respectivos centróides enquanto os quadrados representam as observações. Calculamos os novos centróides e repetimos o processo até a estabilidade ou excedermos o número máximo de iterações.

Capítulo 3

Revisão Bibliográfica

Neste capítulo, discute-se um conjunto de trabalhos que estão relacionados aos temas abordados. O objetivo da Seção 3.1 é sumarizar os diversos trabalhos presentes na literatura promovendo uma visão histórica desde a formalização do problema. As Seções 3.2 e 3.3 estão relacionadas ao método proposto e buscam contextualizar a abordagem utilizada com as várias técnicas de aprendizado de máquina existentes.

Redes neurais têm sido bastante utilizadas em sistemas de QA. Outras aplicações de redes que também tiveram grande sucesso foram em experimentos envolvendo análises de áudio e de imagens. Em particular, destaca-se um ganho considerável quando utilizadas técnicas de transferência de aprendizado e adaptação de domínio. Sistemas de QA também têm aproveitado de técnicas de transferência de aprendizado, porém não foram encontradas pesquisas publicadas que empreguem adaptação de domínio como a proposta, o que sugere que o presente trabalho seja pioneiro nesse aspecto.

3.1 Sistemas gerais de pergunta−resposta

Um dos primeiros trabalhos na área de *Question Answering* foi o de Simmons [Simmons, 1965], analisando 15 programas experimentais que chamou de "*a primeira geração de sistemas de pergunta−resposta*", criados nos 5 anos que precederam o artigo. Esses sistemas incluíam agentes sociais que tentavam derivar informação de conversas e gerar respostas, *front-ends* para repositórios de dados e sistemas que tentavam responder questões em inglês direto de um texto. Simmons chega a uma conclusão otimista afirmando que os conceitos básicos envolvendo um sistema capaz de responder perguntas já são compreendidos e que os anos seguintes serão promissores, mas admite que não espera ver nenhum modelo com utilidade prática em seu futuro próximo. A discussão é encerrada com vários desafios e problemas que devem ser ao menos parcialmente

solucionados para o desenvolvimento de processadores de linguagem de alta qualidade e propósito geral. Após mais de 50 anos, grande parte desses desafios já foi solucionada, mesmo que parcialmente. Existem inúmeros sistemas capazes de responder a perguntas com eficácia considerável dentro de certos cenários. Deve-se ressaltar que a medida que soluções são encontradas, novos obstáculos e ainda mais desafios surgem no contexto de QA.

O primeiro uso em grande escala de métodos de um sistema de pergunta−resposta de amplo domínio foi apresentado em 1999 na conferência *TREC-8* [Voorhees et al., 1999], a oitava edição da *Text REtrieval Conference*. Nos anos seguintes, a trilha de *QA* da *TREC* foi o principal local para testar, validar e discutir novos modelos. Seu relatório contém uma avaliação dos métodos apresentados na conferência no formato de uma competição. Várias perguntas de diferentes assuntos são apresentadas e os programas retornam uma lista de cinco pares *[documento,string]* ordenados (reduzido a somente um par em 2002) com respostas candidatas, que então são avaliadas por juízes humanos. A pontuação ocorre em função da primeira resposta correta encontrada na lista dos pares. Um ponto é atribuído se a *string* do primeiro par contém a resposta correta, $\frac{1}{2}$ se ela se encontra no segundo par, $\frac{1}{3}$ se está presente na terceira e assim sucessivamente. Se os juízes considerarem que nenhum dos pares contêm a resposta, então nenhum ponto é atribuído para o programa nessa pergunta.

Nas notas da aula de Callan [Callan, 2004], vários problemas com o método de avaliação da TREC são levantados:

- Não há penalidades para respostas corretas, mas que não são úteis.
  - "Onde se localiza o Taj Mahal?". Respostas tanto como "Índia"como "Uttar Pradesh"(estado da Índia) ou "Atlantic City"(cidade onde se encontra um cassino chamado *Taj Mahal*) são válidas.
- Não há penalidade para respostas erradas, simplesmente as demais opções de resposta são avaliadas e, no pior caso, o programa deixa de pontuar. Um programa que sempre acerta na segunda resposta consegue a mesma pontuação que um programa que acerta 50% das vezes na primeira resposta e não a encontra nas demais vezes.
- Não há nenhuma recompensa se as respostas apresentadas são complementares.
- Ambiguidade no que é permitido. Por exemplo, encontrar a resposta pela web ao invés do *corpus* fornecido é válido.

O último ano em que a conferência teve uma trilha dedicada à pergunta−resposta foi em 2007 [Dang et al., 2007], com uma única exceção em 2014, na qual foi apresentado uma trilha de *live Question Answering*.

Um dos grandes problemas ao desenvolver sistemas de pergunta−resposta encontra-se na extração de um *corpus* amplo e adequado. Caso os documentos armazenados não correspondam com as consultas realizadas, a eficácia do sistema certamente será prejudicada. Ahn explora o uso da Wikipédia, uma enciclopédia digital e de acesso livre, como *corpus* [Ahn et al., 2004]. Apesar de os resultados obtidos serem decepcionantes, os autores os atribuem aos problemas em sua implementação. Buscaldi procura dividir os termos da Wikipédia em categorias para melhorar seu sistema, porém também adquire resultados desapontadores devido à localidade utilizada em seu experimento (a versão espanhola da Wikipédia e a relativa pequena quantidade de termos)[Buscaldi & Rosso, 2006]. Autores de ambos os trabalhos afirmam que usar a Wikipédia pode levar a resultados significativos se associados com técnicas mais sofisticadas, mas os resultados obtidos levam a crer que as vantagens de se utilizá-la devem ser estudadas cuidadosamente e seu uso deve ser feito com cautela.

Mesmo com um bom *corpus*, ainda podemos encontrar mais problemas na etapa de extração de informação [Pasca & Harabagiu, 2001]. Em particular, os autores destacam que uma das estratégias empregadas por grande parte dos sistemas de pergunta−resposta, assumir que todas as respostas são entidades nomeadas, é uma simplificação exagerada do poder generativo das línguas. Além disso, sistemas tradicionais costumam sofrer com diferenças morfológicas, léxicas ou semânticas entre as palavras do *corpus* e da consulta. Para a pergunta *"When was Berlin's Brandenburger Tor erected?*, muito provavelmente a passagem com a resposta possuirá a palavra *"built"* ao invés de seu hipônimo *erected*, mas não há como garantir que essa relação seja identificada sem auxílio. É então proposto o uso da *WordNet*, uma combinação de dicionário e enciclopédia do inglês, para resolver esses e outros tipos de problema encontrados em modelos de pergunta−resposta e os autores afirmam que seu uso leva a um aumento de 147% na precisão dos sistemas.

Moldovan faz uma sumarização das técnicas de estado da arte na área de pergunta-resposta de amplo domínio e desenvolve um sistema que as utiliza [Moldovan et al., 2003]. As etapas e métodos utilizados desde o processamento da pergunta até a obtenção da resposta final são detalhados. Sua base de testes contém as perguntas utilizadas nas edições passadas da TREC e seus resultados são impressionantes, comparáveis aos melhores competidores de cada edição. Moldovan mostra que o avanço nas técnicas de processamento de linguagem natural tem um impacto direto na qualidade dos sistemas de pergunta−resposta, uma conclusão similar à encontrada

por Harabagiu [Harabagiu et al., 2000], o qual nota que mesmo os métodos recentes mais simples apresentam resultados superiores aos usados anteriormente. Harabagiu consegue um ganho em desempenho de 20% em uma de suas bases de testes quando comparado com técnicas anteriores, atingindo uma acurácia de 84.75%.

A universidade de Stanford desenvolveu recentemente o *Stanford Question Answering Dataset* (SQuAD) [Rajpurkar et al., 2016], uma base de dados de compreensão de leitura. Ela consiste em perguntas propostas por humanos sobre um conjunto de artigos da Wikipedia, cada uma associada a um parágrafo específico. Cada resposta é um segmento de texto que pode ser encontrado em sua respectiva passagem associada. O SQuAD possui mais 100.000 pares de perguntas-respostas ao longo de mais de 500 artigos e é significativamente maior do que os demais conjuntos de dados de compreensão de leitura. Os desafios propostos por essa base preenchem o espaço deixado pelo fim da TREC e seu tamanho possibilita a aplicação de métodos de aprendizagem profunda relatados na literatura como tendo um bom desempenho em problemas afins.

3.2 Redes Neurais aplicadas em pergunta−resposta

Com avanços em aprendizagem profunda de maneira geral, as redes neurais demonstraram ser uma escolha interessante para abordar os mais diversos problemas. Um deles sendo o de compreensão de texto e a modelagem de sistemas de pergunta−resposta. Apesar de requerer uma quantidade maior de dados que os métodos tradicionais de processamento de linguagem natural, métodos baseados em redes neurais frequentemente têm sido associados com uma melhor qualidade nos resultados [Stroh & Mathur, 2016].

Redes convolucionais permitem que o algoritmo possa se concentrar em características espaciais dos dados, na ocorrência de determinados padrões nas diferentes perguntas e sentenças que o sistema possa aproveitar. Redes recorrentes, por sua vez, permitem uma análise temporal concentrando-se na coocorrência de palavras ou padrões em uma determinada ordem ao longo das sentenças. Uma LSTM possibilita que redes neurais recorrentes possam lidar com textos mais longos. Redes de ponteiro (*pointer networks*) não têm a dificuldade das demais redes em necessitar de uma resposta candidata pre-determinada. Elas podem buscar diretamente no texto fonte por possíveis respostas. Mecanismos de atenção e redes de memória permitem que os modelos se concentrem nos fatos mais relevantes para uma determinada pergunta. Estes são só alguns exemplos das vantagens de algumas arquiteturas ao serem aplicadas ao problema proposto. Esta seção foca principalmente em CNNs e LSTMs, uma vez que o modelo proposto no Capítulo 4 é uma combinação dessas duas arquiteturas. Cita-se

os demais tipos de rede, pois idéias pertinentes para trabalhos futuros mencionam o seu uso.

Redes neurais não são capazes de interpretar palavras antes de passarem por uma etapa de pré-processamento. Primeiro, necessitamos convertê-las em valores númericos a serem interpretados. Uma técnica frequentemente empregada é a de *word embeddings*: cada palavra é mapeada para um vetor multidimensional, com o intuito de quantificar o sentido semântico e sintático. Os trabalhos de Bordes [Bordes et al., 2014a, Bordes et al., 2014b] exploram o aprimoramento de técnicas no treino de *embeddings* para melhorar sua performance no cenário específico de pergunta-resposta. Bordes busca aprender uma representação onde os vetores das perguntas e suas respectivas respostas estejam próximos e usa as relações entre entidades presentes na *FREEBASE*[1]. Sua pesquisa é aprofundada ao propor uma nova forma de especialização: adicionar à função objetivo uma matriz para parametrizar a similaridade entre palavras e a resolver usando o algoritmo de otimização L-BFGS[2].

O método de Bordes apresenta algumas desvantagens: é limitado a respostas presentes na FREEBASE[1] e suas relações descritas e o vocabulário empregado é relativamente pequeno, visto que a base contém apenas perguntas e respostas. Esse segundo fator é particularmente relevante. Uma premissa de vários métodos de *embeddings* é que palavras similares apareçam em contextos e sentenças similares. A partir disso, o método adotado pelo *Global Vectors for Word Representation* (GloVe) [Pennington et al., 2014] é utilizar a coocorrência de palavras para quantificar seus vetores. Utilizou-se uma das bases pré-treinadas do GloVe neste trabalho.

Apesar de possuir resultados satisfatórios, a abordagem usando apenas *embeddings* é limitada no sentido de não explorar características do contexto e da sintática das frases. Técnicas de aprendizado profundo podem ser úteis por permitirem uma maior abstração dos dados ao longo das diferentes camadas das redes neurais. Dentre os trabalhos que utilizam CNNs ou LSTMs, têm-se os de Feng [Feng et al., 2015a] e Tan [Tan et al., 2015, Tan et al., 2016], dos quais utilizamos a mesma função objetivo, além do de Saveryn e Moschitti [Severyn & Moschitti, 2015], que não se limita na classificação dos pares pergunta−resposta mas, também, aborda o problema de frase-frase e pergunta-frase.

Saveryn propõe o uso de redes convolucionais para evitar o processo de engenharia de *features* na tarefa de obter a similaridade entre dois segmentos de texto

[1] Uma base de dados prática e escalável de tuplas usada para estruturar o conhecimento humano de maneira geral. Em 2008 possuía mais de 125.000.000 de tuplas, mais de 4000 tipos e mais de 7000 propriedades [Bollacker et al., 2008].

[2] Método de otimização baseado em *hill-climbing* utilizando uma quantidade de memória limitada. Busca minimizar $f(x)$ onde $x \in \mathbb{R}^n$, não possui restrições e f é uma função escalar diferenciavel.

[Severyn & Moschitti, 2015]. No caso específico de QAs, a similaridade entre uma pergunta e sua resposta. Em sua modelagem, Saveryn busca aprender a mapear frases da entrada para vetores, que então podem ser utilizados para calcular sua similaridade. Esse valor é anexado à união dos vetores de ambas as passagens além de algumas caracterísicas adicionais definidas manualmente. Alguns exemplos são a sobreposição de palavras e tamanho das frases. Esse novo documento é então alimentado a uma camada simples de *perceptrons*. A rede proposta não se limita ao problema de pergunta−resposta, podendo ser utilizada em qualquer tipo de relação frase-frase. Em seus experimentos as respostas de cada pergunta são frases completas. Nesse sentido, o problema abordado pode ser visto como o de medir a similaridade pergunta-frase, esta contendo a resposta correta.

Feng explora o uso de uma rede convolucional na tarefa de classificar respostas candidatas para determinadas perguntas [Feng et al., 2015a]. Como diferencial, Feng explora os efeitos do tamanho da camada convolucional. Durante seus experimentos chega a um resultado importante: ao contrário dos usos convencionais de CNNs, no problema de pergunta−resposta é benéfico usar convoluções muito grandes, da ordem de 1000 filtros. Em sua etapa de avaliação, Feng gera sua amostra de respostas candidatas dentre todas as respostas possíveis. Isso difere da abordagem proposta neste trabalho. O conjunto de respostas é restrito ao mesmo assunto, o que implica que a implementação proposta trata de um problema ligeiramente mais difícil. Como as respostas candidatas estão naturalmente relacionadas ao mesmo contexto, estes são casos onde a rede naturalmente teria incerteza.

Como uma contribuição adicional, Feng introduz uma função objetivo diferente das normalmente usadas. Ela busca maximizar a similaridade entre perguntas e suas respectivas respostas corretas, ao passo que também minimiza a similaridade com respostas ruins. No presente trabalho utiliza-se a mesma função objetivo na etapa de treino e otimização da rede (equação 3.1).

$$L = max\{0, M - cos(q, a_{pos}) + cos(q, a_{neg})\} \tag{3.1}$$

- M: margem (constante). Distância mínima que respostas positivas e negativas devem estar entre si.
- q: pergunta depois de ser tratada pela rede.
- a_{pos}: resposta correta depois de ser tratada pela rede.
- a_{neg}: resposta incorreta depois de ser tratada pela rede.

- $cos()$: similaridade de cosseno.

De certa forma, Tan dá continuidade ao trabalho de Feng utilizando a mesma função objetivo e modelando o problema da mesma forma, utilizando porém uma LSTM no lugar da CNN [Tan et al., 2015]. Tan amplia a discussão explorando diferentes arquiteturas, incluindo o uso de dispositivos de atenção, diferentes métodos de *pooling* e acrescentando uma camada de convolução após a camada de LSTM. De maneira geral, estes dois trabalhos mostram que CNNs e LSTMs têm potencial ao tratar de QAs e, inclusive, trabalham bem unidas. No entando, grande parte dos trabalhos que unem essas duas arquiteturas empregam a camada de LSTM após a convolução. Essa arquitetura foi explorada pelos autores em seu trabalho seguinte [Tan et al., 2016] e se mostrou mais efetiva que as demais variantes dos modelos explorados envolvendo CNN e LSTM. Neste trabalho aborda-se essa mesma arquitetura, conseguindo resultados satisfatórios.

Vários métodos já foram testados na própria base do SQuAD e este guarda a classificação dos melhores propostos. Dentre esses trabalhos, foram escolhidos alguns para servirem de *baseline* no Capítulo 5. O primeiro deles é o proposto pelo próprio SQuAD [Rajpurkar et al., 2016]: um modelo de regressão logística que utiliza informações como árvores léxicas e de dependência. Um segundo trabalho utiliza uma GAN (*Generative Adversarial Network*) com componentes discriminativos e generativos para criar questões sintéticas de dados não-rotulados, enriquecendo-se, deste modo, a base de treino [Yang et al., 2017].

Os próximos dois métodos foram propostos como *baselines* para os modelos de seus respectivos artigos. Weissenborn [Weissenborn et al., 2017] detalha o método Neural−BoW, que deriva o tipo léxico de resposta esperado por meio de palavras chave na pergunta (como *who, when, why, how, how ... much*, etc.) ou o primeiro nome após as palavras *Which* ou *What*. O modelo emprega uma rede totalmente conectada na qual as entradas são a concatenação dos *embeddings* das palavra associadas à resposta candidata. O modelo *Chunk-and-Rank* [Yu et al., 2016] processa uma sentença e a pergunta através de redes neurais recorrentes. Em seguida é aplicado um mecanismo de atenção palavra por palavra na frase utilizando a pergunta. O modelo produz representações dos *chunks* e classifica seus *spans* em busca da resposta correta.

3.3 Transferência de Aprendizado

O primeiro trabalho a abordar a transferência de aprendizado foi o de Caruana [Caruana, 1995]. A premissa de seu trabalho se concentra em ser benéfico treinar

redes neurais simultaneamente em tarefas relacionadas. Caruana mostra que o aprendizado numa tarefa qualquer pode ser utilizado nas demais como um *bias*, um ponto de partida. São apresentadas cinco abordagens diferentes para realizar a transferência de aprendizado abordando o problema de reconhecimento de objetos em imagens. Conclui-se que os modelos treinados em várias tarefas têm uma capacidade de generalização maior, são mais eficazes, podem ser treinados com menos iterações e são computacionalmente mais eficientes à medida que o número de tarefas cresce. Caruana também destaca que quanto mais difícil o problema, melhor é o desempenho do modelo multitarefa em comparação com o específico. Este trabalho atraiu grande atenção do universo acadêmico e impulsionou o uso de transferência de aprendizado em redes neurais profundas.

Essa abordagem porém é muito diferente da utilizada neste trabalho. De fato, várias formas diferentes de realizar a transferência de aprendizado foram propostas nos mais de 20 anos desde o trabalho de Caruana. Destaca-se, porém, o de Bengio [Yosinski et al., 2014], no qual este trabalho baseia suas abordagens de transferência de aprendizado. Bengio afirma que ao longo da arquitetura de redes profundas, à medida que aumentamos o grau de abstração da rede, também passamos a analisar características cada vez mais específicas. Isto é, as camadas mais rasas da rede tratam de aspectos gerais dos dados, presentes em diversos tipos de tarefas. As camadas mais profundas, responsáveis pelos maiores níveis de abstração, acabam estando fortemente associadas com aspectos específicos de cada tarefa. Seu trabalho busca quantificar o quão 'transferíveis' são as características das redes e técnicas voltadas para cada profundidade de camadas são exploradas. Relata-se dois fatores que podem ter um impacto negativo na transferência de aprendizado: quando temos camadas co-adaptadas e é realizada a transferência em apenas uma delas e quando camadas superiores se tornam mais especializadas na tarefa original que nas tarefas alvo. De maneira geral, sua conclusão sobre o impacto da transferência de aprendizado é semelhante a de vários trabalhos: utilizar dados distantes da projeção alvo é preferível a inicializar aleatoriamente os pesos das redes. Mostra-se também que a capacidade de generalização de redes é amplamente aumentada quando utilizados pesos transferidos.

Uma das principais premissas em aprendizado de máquina é que as bases de treino e validação devem seguir uma distribuição similar. Esse tipo de cenário muitas vezes não é válido para problemas no mundo-real. A transferência de aprendizado visa remediar esse problema, permitindo que modelos sejam treinados em distribuições ligeiramente diferentes do alvo, sendo posteriormente usados dados relacionados para um ajuste. Um cenário pouco comum de transferência de aprendizado é quando os domínios alvo e de treino são os mesmos, mas as tarefas que devemos realizar sobre

os dados diferem. Desde 1995, várias técnicas foram apresentadas e Pan apresenta um *survey* introduzindo alguns dos conceitos básicos necessários para pesquisadores da área, assim como diversos métodos que emergiram [Pan & Yang, 2010]. De particular relevância para este trabalho, destacam-se as seguintes definições:

Definição 1: *"(Transfer Learning) Dado um domínio fonte DS e uma tarefa de aprendizagem TS, assim como um domínio alvo DT e uma tarefa de aprendizagem TT, a transferência de aprendizagem visa ajudar a melhorar a aprendizagem da função preditiva alvo $fT(.)$ em DT usando o conhecimento em DS e TS, onde $DS \neq DT$, ou $TS \neq TT$."*

Definição 2: *(Transductive Transfer Learning)* Essa definição extende a definição 1, incluindo que uma certa quantidade de dados alvo não-rotulados deve estar presente durante o treino.

A segunda definição aborda a tarefa de adaptação de domínio [Arnold et al., 2007]. No trabalho de Arnold, é explorado o uso de diferentes técnicas relacionadas a SVMs e modelos de entropia. Seu trabalho mostra que mesmo uma quantidade pequena de conhecimento prévio é capaz de levar a um grande aumento na performance de sistemas. Neste trabalho, a diferença entre os domínios está na discrepância das distribuições de probabilidade dos dados fonte e alvo.

Transferência de aprendizado tem sido utilizada em vários problemas distintos nos últimos anos, tendo se mostrado uma metodologia importante. De maneira geral, os trabalhos mostram que é vantajoso realizar uma etapa de pré-treino nas redes usando dados relacionados antes de a especializar nas tarefas alvo. Porém, os trabalhos que exploram o uso de transferência de aprendizado em sistemas de pergunta−resposta são limitados e, portanto, há amplo potencial de pesquisa nessa área.

O trabalho de Bordes é um destes poucos exemplos que abordam a temática de transferência de aprendizado e pergunta−resposta [Bordes et al., 2015b]. Por meio de uma *Memory Network*, busca-se encontrar o fato necessário para responder uma determinada pergunta dentro da memória da rede e das passagens candidatas. Em particular, Bordes cria uma nova base de dados baseada no FREEBASE para atacar esse problema e a combina com outros *benchmarks*. Conclui-se que modelos treinados apenas em uma das bases têm um desempenho ruim nas demais, mesmo quando utilizada a maior delas. Ao realizar o treino da rede em múltiplos tipos de dado, o desempenho sempre aumenta e não foi notado nenhum tipo de 'interação negativa' em seus experimentos. É interessante notar que a proposta deste trabalho difere da de Bordes. Não se deseja buscar mais dados para complementar o treino do modelo, pelo contrário, mostra-se que há benefícios em dividir a base utilizada em sub-domínios, estratificando, portando, os dados.

Abordando tarefas mais genéricas dentro da grande área de processamento de texto, são encontrados mais usos de transferência de aprendizado. Por exemplo, redes convolucionais foram utilizadas na tarefa de classificação de sentenças e aprendizado de múltiplas tarefas [Kim, 2014]. A etapa de *fine-tunning* é realizada sobre os vetores (*embeddings*) do vocabulário da rede, sendo então especializados em cada uma das tarefas abordadas. Diferentes técnicas são analisadas e ganhos de desempenho são relatados.

Quando apenas utilizados os vetores do *Word2Vec* pré-treinados, palavras com relações como antônimos aparecem extremamente próximas, como é o caso do vetor de *good* que tem como vizinho mais próximo *bad*. Um resultado importante de Kim é que, ao realizar a etapa de *fine-tunning*, palavras com sentidos similares se aproximam no espaço vetorial. Um exemplo é o par *good-nice*, que se torna extremamente próximo. Antônimos em contrapartida se distanciam um dos outros. Isso implica que o modelo aprendeu um mapeamento de *embeddings* que representa palavras valorizando seu sentido semântico e não seu papel sintático quando comparamos com o Word2Vec padrão. Isso é uma característica desejável de um QA e, por conta disto, a camada de *embeddings* nunca foi congelada nos experimentos realizados durante o decorrer deste trabalho.

Ainda no âmbito de aprendizado de múltiplas tarefas, também encontramos redes recorrentes sendo utilizadas [Jaech et al., 2016]. O principal objetivo de Jaech é mostrar que, ao empregar conhecimento prévio obtido de outras bases, precisamos de menos dados para ensinar o modelo uma nova tarefa ou domínio. O problema abordado inclui interpretar palavras que não estão presentes no vocabulário. Seus resultados mostram que a combinação de um modelo treinado em múltiplas tarefas com um vocabulário aberto aumenta a capacidade de generalização do modelo.

Todavia, o grande sucesso de transferência de aprendizado está associado ao seu uso na análise de imagens e áudio. Recentemente, Marczewski teve ótimos resultados ao se aproveitar de adaptação de domínio em uma rede que combina camadas convolucionais e recorrentes na tarefa de detecção de emoções em faixas de áudio [Marczewski et al., 2017]. Suas abordagens de transferência de aprendizado são similares às deste trabalho no sentido de alternar camadas de sua rede para serem especializadas. Similarmente às conclusões obtidas neste trabalho, observa-se que certas abordagens têm afinidade maior com certos domínios. Uma segunda conclusão é que nem sempre é benéfico utilizar todos os dados presentes. Como seu conjunto de treino foi criado a partir da união de várias bases menores, alguns de seus domínios diferem muito dos demais. Certas tarefas são prejudicadas pela presença dessas instâncias tão divergentes durante a transferência de aprendizado.

Capítulo 4

Implementação

Nesse capítulo são tratados os detalhes de implementação do método desenvolvido. Primeiramente, é descrita a arquitetura da rede proposta, similar à do trabalho de Tan e as diferentes abordagens de transferência de aprendizado que buscam aproveitar dessa arquitetura. Em seguida, é descrita umas das principais contribuições deste trabalho: aproveitar a informação presente no QA pergunta-sentença para melhorar a performance do QA pergunta-*span*. Finalmente, é explicada a metodologia básica por trás da divisão dos domínios do SQuAD, etapa essencial para a transferência de aprendizado.

Todas as redes avaliadas (CNN, LSTM, RNN, Resnet, MLP e o modelo proposto) foram implementadas em *python* por meio pacote *Keras* na versão 1.2.2 e usando o *backend* Theano versão 0.9.0. As implementações do Doc2Vec e K-means utilizadas são as dos pacotes NLTK 3.2.2 e Gensim 1.0.1 respectivamente.

4.1 Arquitetura proposta

O modelo de QA proposto é formulado como uma função $f(q, a; \theta)$ parametrizada por θ que mapeia um par de pergunta−resposta para uma pontuação de relevância. Dada uma pergunta q e uma lista de respostas candidatas $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, $f(q, a; \theta)$ é usado para calcular a relevância entre q e cada candidato $a_i \in \mathcal{A}$. A resposta mais altamente classificada é retornada como sendo a correta. É assumido um cenário de aprendizagem em que as questões são mapeadas para um conjunto de domínios, divididos pelos seus assuntos. Isso permite aprender modelos específicos $f^d(q, a; \theta)$ para cada domínio d. Nesse caso, os parâmetros θ são encontrados maximizando-se a relevância da questão-resposta presentes em um domínio e tópico específico. Utiliza-se a mesma formulação tanto para modelos no nível de sentenças quanto de *spans*.

É empregada uma arquitetura CNN-biLSTM que é semelhante à proposta por Tan [Tan et al., 2016]. Na camada mais rasa, o componente convolucional enfatiza as interações locais de $n-$gramas. Nas mais profundas, a camada recorrente é capaz de capturar as dependências de longo alcance com base na convolução dos $n-$gramas, também sendo capaz de filtrar e ignorar informações locais de pouca importância. Essa arquitetura complementar garante ao modelo a capacidade de avaliar tanto características espaciais quanto temporais dos dados.

A figura 4.1 ilustra a arquitetura proposta. Tanto a pergunta quanto a resposta passam pela mesma camada de *embedding* de dimensão 100 antes de serem avaliadas no restante da rede. Utiliza-se apenas uma camada convolucional e os experimentos exploram o uso de 1000 e 2000 filtros, divididos entre os tamanhos 2,3,5 e 7. A saída da CNN é alimentada numa LSTM bidirecional com 141 neurônios. Em seguida, realiza-se *maxpooling* na rede com uma *pool* de tamanho 3. As respectivas saídas são avaliadas em uma camada totalmente conectada de dimensão 300 que utiliza como ativação uma função tangente. Como etapa final, compara-se a distância de cosseno entre as representaçõs da pergunta e da resposta, obtendo o quão similar o modelo acredita que sejam.

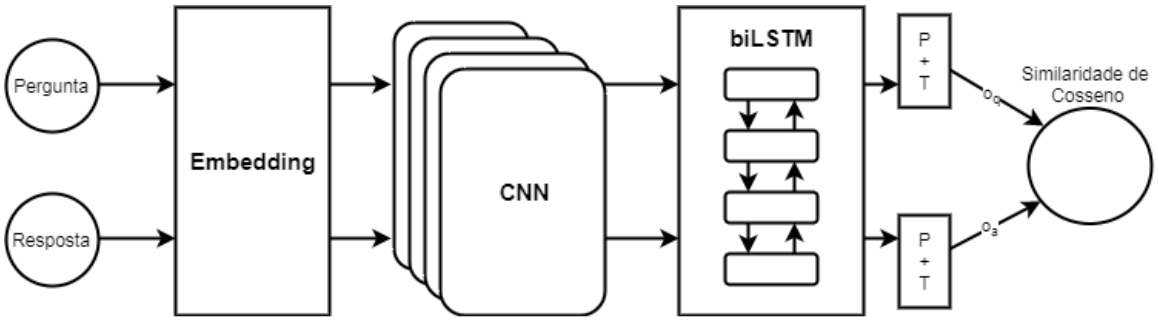


Figura 4.1: Arquitetura da rede proposta. A pergunta e a resposta são processadas por uma CNN-biLSTM e mede-se a similaridade de cosseno entre elas.

Especificamente, as palavras são representadas como um vetor de baixa dimensionalidade [Mikolov et al., 2013, Pennington et al., 2014]. A camada da CNN recebe como entrada uma sentença no formato de uma matriz $D \in \mathcal{R}^{kE \times L}$, onde cada coluna l em D consiste na concatenação de k vetores de tamanho E, centrados na l^{a} palavra e L limita o tamanho da frase, como ilustrado pela Figura 4.2. A CNN aplica c filtros, resultando em uma matriz $\mathbf{X} \in \mathcal{R}^{c \times L}$ tal que:

$$\mathbf{X} = \tanh(\mathbf{W}D) \tag{4.1}$$

onde $\mathbf{W}$ são os parâmetros de convolução. Uma diferença importante do trabalho de

Tan é que são aplicados filtros de diferentes tamanhos. A estrutura biLSTM recebe a matriz $\mathbf{X}$ como entrada e, em seguida, utiliza-se *max-pooling* nos vetores de saída da biLSTM para obter as representações de q e a da pergunta e resposta respectivamente. As seções da rede que tratam da pergunta e da resposta têm seus pesos compartilhados. Foi mostrado que isto leva a uma convergência mais rápida e um melhor desempenho [Feng et al., 2015a, Tan et al., 2015, Tan et al., 2016].

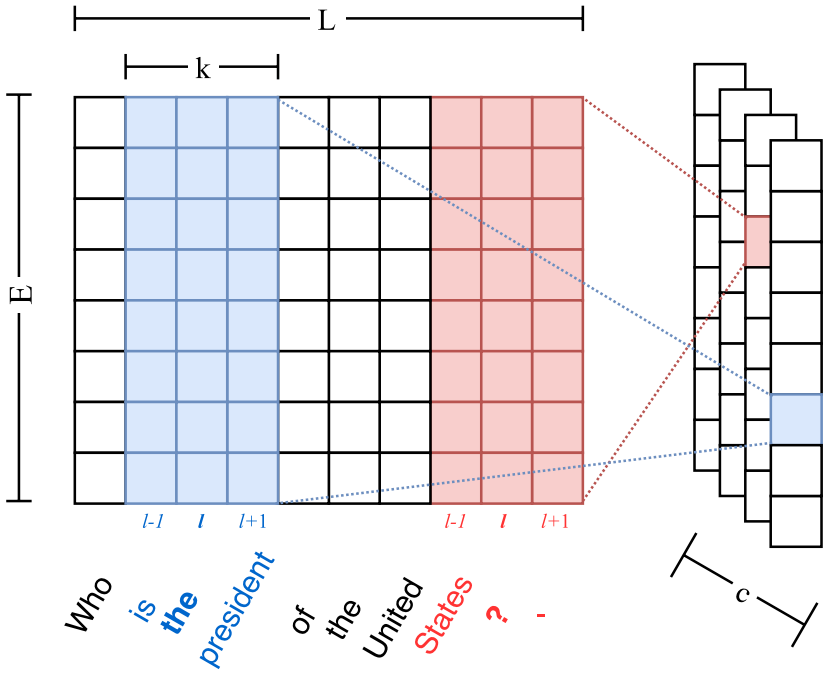


Figura 4.2: Arquitetura de uma camada convolucional de uma dimensão que recebe como entrada os *embeddings* das palavras de uma sentença. E representa o tamanho dos *embeddings*, enquanto L e k representam respectivamente o tamanho da sentença e do campo receptivo avaliado pelos c filtros, centrados sempre na l-ésima palavra.

Os parâmetros θ do modelo são obtidos ao treiná-lo como em um problema de clasificação par-a-par. Semelhante a Feng, Weston e Hu [Feng et al., 2015a, Weston et al., 2014, Hu et al., 2014], definimos a função objetivo como uma *hinge-loss* com base na semelhança de cosseno:

$$\mathcal{L} = \max\{0, M - \cos(q, a_+) + \cos(q, a_-)\} \tag{4.2}$$

onde a_+ é a resposta correta, a_- é uma resposta incorreta escolhida aleatoriamente de um grupo de respostas candidatas, e M é a margem, que simboliza a distância mínima desejada entre (q, a_+) e (q, a_-) . Durante o treino, escolhe-se K amostras aleatórias de respostas negativas para cada pergunta, sendo utilizada apenas a de erro $\mathcal{L}$ mais alto

para atualizar o modelo. Finalmente, é calculada a similaridade de cosseno entre as representações geradas.

4.2 Transferência de aprendizado

Assume-se a presença de poucos pares de pergunta−resposta nos domínios alvo. Portanto, uma adaptação direta é propensa à sobre-ajuste. Também assume-se que o conjunto de treino original é composto por pares de pergunta−resposta pertencentes a diferentes domínios. Assim, o objetivo é treinar um modelo de QA multi-domínio que seja capaz de classificar respostas corretas mais alto que as incorretas. Certamente características específicas de cada domínio são mais adequadas para processar os dados, porém é dificil aprendê-las com uma base pequena. Existem ainda alguns padrões mais simples e gerais, que estão presentes em vários domínios. Exemplos dessas características simples de baixo nível podem incluir a coocorrência de $n-$gramas, enquanto exemplos de padrões de alto nível podem incluir sequências específicas de $n-$gramas. Tendo isso em vista, são propostas três abordagens distintas de transferência de aprendizado.

A principal intuição explorada para transferibilidade é que os padrões e características analisados pelos modelos devem eventualmente transitar de gerais para específicos ao longo da arquitetura de rede, diminuindo significativamente com o aumento da discrepância de domínio [Yosinski et al., 2014]. Em outras palavras, camadas mais profundas estão altamente relacionadas com os domínios específicos e a discrepância entre os domínios de treino e de avaliação as afetam negativamente. Camadas mais rasas, por sua vez, sofrem uma menor influência dos domínios específicos de treino por tratarem de características mais gerais e presentes em grande parte da base de dados. Porém, uma vez que estamos lidando com vários domínios simultaneamente, também considerou-se múltiplas abordagens de transferência com a esperança que algumas se sobressaissem em cenários diferentes. São elas:

T1: Nenhuma camada é mantida congelada durante a etapa de *fine-tunning*, o que significa que os erros são propagados por toda a rede ao atualizar os pesos.

T2: Apenas a camada convolucional é mantida congelada durante a etapa de *fine-tunning*.

T3: Apenas a camada convolucional é mantida congelada durante a etapa de *fine-tunning*. Porém, a camada recorrente tem seus pesos inicializados aleatoriamente.

Antes da especialização, sempre realizou-se um pré-treino na rede onde todas as amostras de entrada são usadas independentemente do domínio. Esta etapa é essencial para permitir que os modelos aprendam características gerais comuns. Isto também permite que os modelos mantenham alguma memória de outros domínios, mesmo quando especializados, permitindo, portanto, que mantenham um desempenho razoável quando avaliados em domínios diferentes do alvo.

Nas abordagens T2 e T3, assume-se que a etapa de pré-treino é suficiente para que o modelo aprenda características gerais. A etapa de especialização se dedica a aprender características específicas de alto nível. A abordagem T3 enfatiza essa hipótese, descartando o que foi aprendido em alto nível nos dados gerais. Apesar disso, reconhece-se que podem existir algumas características de baixo-nível pertinentes aos domínios específicos. A partir disso, foi proposto T1, no qual a camada mais rasa também é atualizada. O Capítulo 5 explora comparativamente T1, T2 e T3 e confirma que certas abordagens têm mais sucesso em certos domínios que as demais, não existindo uma superior na média.

4.3 Condicionando informação das sentenças no modelo padrão

Foram treinados dois modelos de QA separados usando a mesma arquitetura CNN−biLSTM. Definiu-se o modelo de QA sobre os *spans* para um domínio d como sendo $f^d_{span}(q, a; \theta)$, e o modelo de QA ao nível das sentenças para um domínio d como $f^d_{sentence}(q, s; \theta)$. Em contraste com as abordagens típicas que condicionam a resposta às suas frases de origem, treinando uma rede única que recebe toda a informação como entrada [Sultan et al., 2016, Lee et al., 2016], os modelos propostos são treinados de maneira independente, usando a mesma base de treino, mas com um pequeno pré-processamento.

No modelo que trata as sentenças, temos o mesmo conjunto de perguntas que o modelo baseado nos *spans*, mas as respostas são substituídas por suas frases de origem. O que se propõe é explorar a informação do contexto em tempo de teste, na forma de sentenças. Dada uma pergunta q, soma-se a relevância de uma resposta a_x com a relevância da sentença s_x para esta pergunta, dado que a_x foi extraída de s_x. Temos que a resposta a retornada pelo modelo combinado pode ser expressada por:

$$\text{argmax}_a \quad [f^d_{span}(q, a; \theta) + f^d_{sentence}(q, s; \theta)] \quad | \quad a \subseteq s \tag{4.3}$$

Ou seja, procura-se a resposta a que maximize $f^d_{span}(q, a; \theta) + f^d_{sentence}(q, s; \theta)$. A intuição básica é que frases relevantes podem fornecer informações úteis para escolher a resposta correta no nível de *spans*. Suponha que a^+ seja a resposta correta para uma pergunta arbitrária q. Suponha também que $f^d_{span}(q, a^+; \theta) \approx f^d_{span}(q, a^-; \theta)$. Nesse caso, se uma frase s^+ contendo a^+ for classificada acima de outra frase s^- contendo a^-, a equação 4.3 aumentará as chances de a^+ ser classificado acima de a^-.

Alguns dos experimentos relatados no Capítulo 5 validam essa hipótese. Um exemplo ajuda a entender os motivos que levam ao aumento na performance do QA de *spans* ao utilizar a informação de outro QA distinto. Seja a pergunta *'Qual equipe ganhou a última Liga dos Campeões da UEFA?'*. Ambas *'Barcelona'* e *'Real Madrid'* são respostas candidatas adequadas. No nível de *spans*, identificar a resposta correta pode ser uma tarefa difícil, mas se consideradas as frases de onde estas respostas foram retiradas, o problema se torna trivial. Essas respostas foram extraídas das respectivas sentenças:

- *"A vitória na final resultou no* ***Real Madrid*** *sendo o primeiro time a defender com sucesso seu título na era da UEFA Champions League."*
- *"****Barcelona*** *é um dos principais centros turísticos, econômicos, comerciais e culturais do mundo."*

O problema de classificar a relevância entre perguntas e sentenças é mais fácil que o dos *spans*. Além de existirem mais palavras nas passagens candidatas, geralmente elas têm algum nível de sobreposição de $n-$gramas com a pergunta. Isso faz com que os modelos de QA sobre as sentenças tenham uma performance consideravelmente superior. Assim sendo, dificilmente haveria uma interação negativa ao adicionar esse tipo de informação no problema dos *spans*.

4.4 Base de dados e divisão dos domínios

O *Stanford Question Answering Dataset* (SQuAD) [Rajpurkar et al., 2016] é uma base de dados de compreensão de leitura desenvolvida em 2016. Ela consiste em perguntas propostas por humanos sobre um conjunto de artigos da Wikipédia, cada uma associada a um parágrafo específico. Cada resposta é um segmento de texto que pode ser encontrado em sua respectiva passagem associada. A Figura 4.3 ilustra um exemplo de um conjunto de algumas perguntas. O SQuAD possui mais 100.000 pares de perguntas-respostas ao longo de mais de 500 artigos e é significativamente maior do que os conjuntos de dados de compreensão de leitura anteriores.

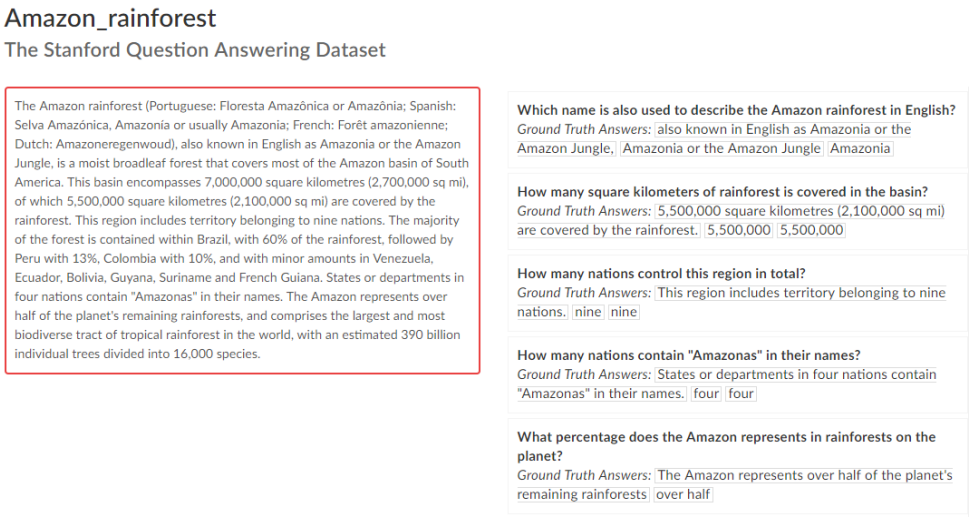


Figura 4.3: Exemplo de um dos parágrafos e suas respectivas perguntas contidas na base do SQuAD. Para cada questão são apresentadas três possibilidades respostas que constituem um segmento do parágrafo, embora nem sempre elas sejam distintas entre si.

Cada artigo extraído da Wikipedia é dividido em seções na forma de parágrafos. Para cada seção, existe um conjunto de perguntas cujas respostas são um pequeno fragmento de texto. Durante sua criação, foi sugerido que, quando possível, se evitasse utilizar palavras encontradas no parágrado na confecção das perguntas. Cada pergunta possuí até três respostas corretas, não necessariamente distintas.

Essa base não é dividida em assuntos. As únicas divisões presentes são em artigos e parágrafos. Uma das propostas deste trabalho é mostrar que não é necessário utilizar dados adicionais para realizar a adaptação de domínio em casos similares. Para isso, dividiu-se a base do SQuAD em diferentes tópicos manualmente. Essa tarefa pode ser descrita em três simples passos.

Para cada um dos 446 artigos presentes na base de treino, são atribuídos um rótulo referente à sua temática dentre um grupo de 50 dos 1000 principais tópicos de acordo com a Wikipédia. Essa divisão acabou levando a domínios muito pequenos, com apenas um ou dois artigos. A segunda etapa consiste em remediar esse problema. Tópicos com temática similar foram unidos em um único macro-tópico, com o objetivo de nenhum domínio específico ter menos do que dez artigos. Esse é o caso do domínio ciência, que é composto pelas temáticas física, química e astronomia. Note que o domínio de Biologia também possui um tema similar. Por ser grande o suficiente, ele foi mantido como um domínio único. Isso faz com que exista uma maior diversidade de assuntos para a etapa de transferência de aprendizado. O mesmo ocorre com os domínios cidade, país e estado, por exemplo.

Alguns artigos se encaixam em mais de uma temática abordada. Nesses casos específicos, ambos os domínios contêm as perguntas referentes a este artigo. Apenas sete casos foram encontrados: os artigos referentes a Madona e Steven Spielberg (pessoa e entretenimento), os referentes aos papas Paulo VI e João XXIII (pessoa e religião), sobre a comercialização de energia renovável (legislação e tecnologia) e os referentes à religião na Roma antiga e sobre a separação da igreja e estado nos EUA (religião e história). Assim, 17 domínios foram confecionados para este trabalho. A última etapa consistiu em rotular os tópicos da base de avaliação em função dos obtidos na etapa anterior. A tabela 4.1 ilustra a divisão proposta da base do SQuAD.

Tabela 4.1: Tamanho das bases de treino e avaliação após a divisão manual dos domínios.

	Treino		Avaliação	
	N. artigos	N. perguntas	N. artigos	N. perguntas
Person	48	12399	5	4697
History	38	10060	5	3607
Country	43	8402	-	-
City	29	7784	3	1478
Entertainment	33	6479	2	1284
Biology	37	6351	4	2480
Location	36	6139	5	3334
Technology	32	5094	4	2814
Law	27	3840	5	3592
Religion	18	3788	2	1168
Sports	16	3478	2	3172
Organization	17	2854	1	648
Thing	16	2845	1	294
Education	16	2527	5	2053
State	11	2527	2	1062
Language	16	2304	-	-
Science	13	2128	5	4356
Total	**446**	**88999**	**51**	**36039**

4.5 K-means

Explorou-se também um segundo método de extrair os domínios, porém de maneira automatizada. Inicialmente foi criada uma representação dos parágrafos dos artigos usando Doc2Vec [Le & Mikolov, 2014]. Foram avaliados três métodos para isso:

- **E1**: Modelo pré-treinado baseado em DBOW e usando toda a Wikipedia [Lau & Baldwin, 2016].

- **E2**: Modelo que foi treinado diretamente nos artigos do SQuAD. Utiliza uma janela de 5 palavras ignorando *stopwords* e gera uma representação de dimensionalidade 100.

- **E3**: Modelo que foi treinado diretamente nos artigos do SQuAD. Utiliza uma janela de 15 palavras ignorando *stopwords* e gera uma representação de dimensionalidade 100.

A partir desta representação vetorial, foi utilizado o algoritmo K−means buscando agrupar parágrafos com vetores próximos em um único domínio. Foi avaliado o desempenho dos sistemas de QA com cinco e dezessete *clusters*. Espera-se que, ao utilizar os *embeddings* gerados pelo Doc2Vec e o método de clusterização K-means, a divisão de domínios seja baseada tanto no vocabulário dos parágrafos quanto em sua temática.

Uma vantagem dessa abordagem é que os domínios possuem tamanhos equilibrados devido à natureza do algoritmo K−means de minimizar a entropria total do sistema. A segunda é que é possível realizar uma divisão de grão-fino, baseado diretamente nos parágrafos. Considere, por exemplo, um artigo sobre a cidade de Paris. Alguns parágrafos podem tratar de artistas famosos como Manet ou Picasso, enquanto outros podem abordar aspectos geoeconômicos. Estes parágrafos relacionados a artistas certamente são extremamente similares a temas que tratam do Modernismo. A informação sobre a economia de Paris, provavelmente, é menos útil para o modelo que a relacionada à arte.

Uma terceira vantagem é limitada apenas ao método com cinco *clusters* de parágrafos, no qual os dados se encontram menos extratificados. Isso implica não somente em bases maiores como também na necessidade de treinar menos modelos. Utilizar dezessete *clusters*, por sua vez, nos permite ter uma comparação direta com a rotulagem manual. As tabelas 4.3 e 4.4 resumem as estatísticas dos domínios gerados a partir do Doc2Vec + K−means. A Tabela 4.2 ilustra as estatísticas dos domínios rotulados manualmente e é possível observar que o método automático gera divisões mais estáveis.

Tabela 4.2: Estatísticas dos domínios criados rotulando manualmente. Tanto as bases de treino e avaliação apresentam um alto desvio percentual, indicando que o tamanho dos domínios está altamente desbalanceado. Isto pode ser observado pela descrepancia das maiores e menores bases presentes ilustrados pelos valores em *Max* e *Min*.

	Treino		Avaliação	
Rotulado	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	2701	12399	292	4697
Min	459	2128	18	294
Desvio	*57.1%*	*57.7%*	*56.3%*	*57.4%*
Total	**19348**	**88999**	**2143**	**36039**
Média	***1138.1***	***5235.3***	***126.0***	***2119.9***

Tabela 4.3: Estatísticas dos domínios criados em cada método para cinco *clusters*. Os valores de desvio percentual são extremamente menores que os do método de divisão manual, indicando bases muito mais estáveis em relacão ao seu tamanho. Todavia, isto é esperado dado a presença de menos divisões.

	Treino		Avaliação	
E1	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	4090	19652	540	9627
Min	3303	14583	309	4966
Desvio	*8.2%*	*10.3%*	*22.0%*	*26.8%*

	Treino		Avaliação	
E2	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	4454	21112	489	8386
Min	3139	15644	315	5196
Desvio	*12.2%*	*11.9%*	*15.1%*	*18.4%*

	Treino		Avaliação	
E3	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	4685	21286	468	8090
Min	2788	14121	364	5561
Desvio	*18.1%*	*16.6%*	*10.0%*	*11.6%*
Total	**18896**	**87599**	**2067**	**34726**
Média	***3779.2***	***17519.8***	***413.4***	***6945.2***

Tabela 4.4: Estatísticas dos domínios criados em cada método para dezessete *clusters*. Os valores de desvio percentual são menores que os do método de divisão manual, indicando bases mais estáveis em relacão ao seu tamanho, o que ilustra uma das vantagens da divisão automática de domínios.

	Treino		Avaliação	
E1	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	1687	7715	191	3644
Min	615	3010	27	451
Desvio	*24.2%*	*25.1%*	*43.5%*	*40.9%*

	Treino		Avaliação	
E2	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	1974	7921	212	3447
Min	483	2241	32	642
Desvio	*29.9%*	*25.8%*	*39.8%*	*41.9%*

	Treino		Avaliação	
E3	N. parágrafos	N. perguntas	N. parágrafos	N. perguntas
Max	2126	8461	301	4569
Min	611	3020	53	886
Desvio	*31.6%*	*26.9%*	*48.3%*	*44.0%*

Total	**18896**	**87599**	**2067**	**34726**
Média	***1111.5***	***5152.9***	***121.6***	***2042.7***

Capítulo 5

Experimentos

Neste capítulo serão discutidos os procedimentos de avaliação utilizados e relatados os resultados do modelo multi-domínio utilizado, referido como CNN−biLSTM−DA. Em particular, os experimentos visam responder as seguintes questões de pesquisa (QP):

QP1: Qual a relação entre os diferentes domínios? As bases de treino e teste são similares?

QP2: Redes convolucionais e recorrentes são adequadas para tratar do problema de respostas a perguntas?

QP3: A adaptação do domínio melhora a eficácia dos nossos modelos CNN−biLSTM para o QA envolvendo *spans*?

QP4: Qual abordagem de transferência de aprendizado é mais apropriada para cada domínio avaliado?

QP5: As informações no nível da sentença melhoram o desempenho do QA em nível de *spans*?

QP6: Qual o impacto da aplicação de métodos simples de identificação de tópicos?

QP7: Como nossos modelos CNN−biLSTM se comparam aos modelos existentes?

Para isso, foram comparados os resultados dos nossos modelos com vários métodos desenvolvidos em outros trabalhos e executados no SQuaD. A descrição de cada um deles pode ser encontrada no Capítulo 3. Como uma base fraca também considerou-se algumas variantes simples do modelo empregado que não utilizam adaptação de domínio. A seguir os *baselines* utilizados:

- Modelo com nenhuma transferência (CNN−biLSTM−NT): o modelo proposto é treinado usando todos os domínios base e nenhum *finetunning* é aplicado.
- Modelo treinado em domínios específicos (CNN−biLSTM−DS): o modelo proposto é treinado diretamente no domínio alvo, sem uma etapa de pré-treino.
- [Rajpurkar et al., 2016]: um modelo de regressão logística.
- [Yang et al., 2017]: um modelo que utiliza uma GAN para criar questões sintéticas.
- [Weissenborn et al., 2017]: Neural−BoW emprega uma rede totalmente conectada na qual as entradas são a concatenação de *embeddings* relacionados à resposta candidata.
- [Yu et al., 2016]: Chunk-and-Rank utiliza redes recorrentes e mecanismos de atenção para criar representações de passagens candidatas e classificá-las.

Em todos os experimentos, os modelos avaliados classificam 19 respostas candidatas de acordo com a semelhança de cosseno com a questão dada. Essa amostra é composta pela alternativa correta e outras 18 respostas associadas às demais perguntas no mesmo parágrafo. A medida usada para avaliar a efetividade de nossos modelos é a Correspondência Exata (EM, *Exact Match*) [Rajpurkar et al., 2016]. Além de avaliar se a resposta correta é a melhor classificada entre a lista de candidatas ($EM@1$), também é considerado se esta está entre as 5 melhores classificadas ($EM@5$) como uma métrica mais relaxada. Os resultados relatados são a média de cinco iterações e são usados para avaliar o desempenho geral dos modelos. Para assegurar a relevância dos resultados, utilizamos o teste-t pareado com um p−value ≤ 0.05 para garantir uma significância estatística [Sakai, 2014].

5.1 Relação entre os domínios

Inicialmente, desejamos responder QP1, relacionado com a estrutura da base de dados e a divisão de domínios proposta. Para isso, utilizou-se a divergência de KL aplicada a textos. Ela nos permite observar o quão similares duas distribuições são. Note, porém, que ela é uma métrica não simétrica. Isso implica que podemos ter cenários onde concluímos que uma base A qualquer é similar a B, mas que a recíproca não é necessariamente verdade.

A Figura 5.1 ilustra a divergência de Kullback–Leibler entre os domínios da base de dados avaliada. Valores em branco representam uma menor divergência e implicam

que os domínios são próximos. Quanto maior a intensidade de cinza, maior a divergência e, portanto, menor a similaridade. Os maiores domínios possuem uma divergência menor. Isso está associado ao fato de possuírem um vocabulário mais abrangente já que são mais extensos. A Figura 5.2 mostra a divergência de KL quando realizamos esse experimento usando apenas as 2000 palavras mais frequentes de cada domínio. Nesse segundo cenário, observamos correlações claras como entre cidades e países.

Analisou-se também a divergência entre as bases de teste e avaliação (Figura 5.3). Idealmente, os valores nas diagonais deveriam não somente ser os menores de cada linha, mas pequenos de maneira geral. Apesar de nem sempre termos os menores valores presentes na diagonal, eles ainda são pequenos e muito próximos do ideal. A exceção deste padrão ocorre principalmente nos domínios *Organization*, *Sports* e *Thing*, que, inclusive, apresentam alguns dos maiores valores de divergência KL. Esse fenômeno tem um impacto significativo principalmente nos experimentos da Seção 5.4.

TAMANHO	P \ Q	PERSON	HIST.	CNT.	CITY	ENT.	BIO.	LOC.	TECH.	LAW	RLG.	SPORTS	ORG.	THING	SCHOOL	STATE	LNG.	SCI.
12399	PERSON	0.00	3.82	4.24	4.69	4.85	5.96	4.60	6.26	5.24	5.43	5.93	5.68	6.38	5.85	6.24	6.97	7.03
10060	HISTORY	3.14	0.00	3.37	4.18	4.91	5.56	3.88	5.72	4.39	4.80	5.60	5.13	5.53	5.72	5.50	6.07	6.35
8402	COUNTRY	3.37	3.28	0.00	4.05	5.18	5.57	3.78	5.88	4.45	5.28	5.75	5.20	5.82	5.65	5.31	5.87	6.54
7784	CITY	3.90	4.18	3.96	0.00	4.68	6.27	3.74	5.87	5.59	6.43	5.20	5.85	6.22	5.55	4.47	6.90	7.02
6479	ENTERTAINMENT	4.06	5.13	5.26	5.02	0.00	6.26	5.46	5.25	6.01	6.93	5.79	6.43	6.55	6.28	6.80	7.49	7.21
6351	BIOLOGY	4.88	5.23	5.48	6.14	5.74	0.00	5.82	6.19	5.78	6.68	7.27	6.61	5.96	6.95	7.38	7.08	5.95
6139	LOCATION	3.90	3.87	3.85	3.88	5.17	5.94	0.00	6.16	5.41	6.04	5.82	5.88	5.85	5.90	5.33	6.65	6.80
5094	TECHNOLOGY	5.39	5.40	5.86	5.93	4.75	5.91	6.08	0.00	5.72	7.67	6.77	6.52	6.50	6.78	7.49	7.69	6.31
3840	LAW	3.54	3.56	3.71	4.85	5.10	5.20	4.62	5.27	0.00	5.32	5.98	4.95	5.98	5.73	6.05	6.33	6.22
3788	RELIGION	3.64	3.86	4.65	5.52	5.97	6.16	5.43	7.33	5.33	0.00	7.26	6.49	6.80	6.47	7.12	6.81	6.93
3478	SPORTS	4.75	4.99	5.01	4.59	4.97	7.17	5.32	6.21	6.23	7.80	0.00	6.31	7.13	6.28	6.44	8.30	8.17
2854	ORGANIZATION	3.93	4.00	4.22	4.79	5.14	5.70	4.85	5.55	4.80	6.36	5.73	0.00	6.45	5.80	6.28	7.33	6.96
2845	THING	5.03	4.89	5.27	5.70	5.92	5.66	5.33	6.00	6.16	7.16	7.01	7.09	0.00	7.05	6.95	7.72	6.61
2527	SCHOOL	3.96	4.50	4.49	4.45	5.08	6.03	4.54	6.30	5.26	6.14	6.02	5.55	6.91	0.00	6.00	7.09	6.99
2527	STATE	4.26	4.40	4.18	3.47	5.45	6.52	4.00	6.37	5.83	6.67	5.77	6.12	6.49	6.16	0.00	7.13	7.51
2304	LANGUAGE	4.73	4.64	4.43	5.60	5.85	5.88	5.38	6.40	5.69	5.85	6.87	6.75	6.62	6.67	7.11	0.00	7.10
2128	SCIENCE	5.19	5.18	5.50	6.11	5.96	5.00	5.85	5.23	5.76	7.01	7.66	7.17	6.16	7.07	7.33	7.81	0.00
88999	*Total*	*67.67*	*70.92*	*73.47*	*78.98*	*84.72*	*94.80*	*78.67*	*96.01*	*87.65*	*101.54*	*100.43*	*97.74*	*101.36*	*99.89*	*101.81*	*113.22*	*109.70*

Figura 5.1: Divergência de Kullback–Leibler entre os domínios presentes na base de dados.

TAMANHO	P \ Q	PERSON	HIST.	CNT.	CITY	ENT.	BIO.	LOC.	TECH.	LAW	RLG.	SPORTS	ORG.	THING	SCHOOL	STATE	LNG.	SCI.
12399	PERSON	0.00	2.86	3.08	3.28	3.43	4.40	3.06	4.51	3.34	3.45	3.74	3.53	4.12	3.51	3.67	4.38	4.51
10060	HISTORY	2.33	0.00	2.10	2.81	3.62	4.00	2.33	4.26	2.74	3.06	3.55	3.02	3.40	3.48	3.09	3.56	3.90
8402	COUNTRY	2.78	2.42	0.00	2.92	4.05	4.26	2.51	4.60	3.10	3.58	4.00	3.35	3.87	3.71	3.02	3.62	4.34
7784	CITY	3.64	3.76	3.27	0.00	3.88	5.54	2.80	4.98	4.52	5.08	4.03	4.37	4.84	4.03	2.78	5.04	5.20
6479	ENTERTAINMENT	3.69	4.43	4.40	3.96	0.00	5.17	4.30	4.21	4.49	5.21	3.98	4.71	4.74	4.30	4.66	5.15	5.12
6351	BIOLOGY	4.44	4.43	4.32	4.84	4.71	0.00	4.48	4.66	4.25	4.61	5.33	4.70	3.99	4.70	5.03	4.58	3.92
6139	LOCATION	3.16	2.93	2.83	2.55	3.96	4.55	0.00	4.78	3.75	4.22	4.00	3.82	3.95	3.79	3.08	4.31	4.62
5094	TECHNOLOGY	5.21	4.91	5.02	5.00	4.12	4.85	5.09	0.00	4.57	5.87	5.01	4.87	4.93	5.19	5.44	5.53	4.59
3840	LAW	3.36	3.26	3.07	4.00	4.48	4.41	3.73	4.49	0.00	3.94	4.59	3.64	4.36	4.21	4.13	4.46	4.46
3788	RELIGION	3.55	3.72	4.27	4.94	5.32	5.22	4.61	6.19	4.30	0.00	5.76	5.00	5.18	4.87	5.14	4.95	5.09
3478	SPORTS	5.19	5.35	4.81	4.42	4.52	6.62	4.86	5.71	5.75	6.61	0.00	5.60	6.02	5.08	5.08	6.64	6.61
2854	ORGANIZATION	3.94	3.89	3.94	4.20	4.70	5.31	4.21	5.02	4.04	5.18	4.63	0.00	5.18	4.41	4.60	5.55	5.39
2845	THING	4.94	4.78	4.78	4.88	5.24	4.75	4.55	5.09	4.93	5.42	5.49	5.42	0.00	5.39	5.18	5.46	4.83
2527	SCHOOL	4.51	4.99	4.69	4.38	5.26	6.01	4.30	6.31	5.19	5.79	5.82	5.26	6.34	0.00	4.89	6.02	6.15
2527	STATE	4.61	4.68	4.00	3.24	5.34	6.31	3.72	6.05	5.26	5.95	5.18	5.22	5.60	5.07	0.00	5.82	6.28
2304	LANGUAGE	5.23	4.92	4.50	5.65	6.17	5.79	5.06	6.27	5.23	5.03	6.37	5.88	5.60	5.96	5.84	0.00	6.11
2128	SCIENCE	5.69	5.35	5.36	5.62	5.79	4.82	5.68	4.75	4.98	5.82	6.43	5.95	5.13	5.84	5.89	6.02	0.00
88999	*Total*	*66.28*	*66.67*	*64.44*	*66.69*	*74.56*	*81.99*	*65.30*	*81.88*	*70.44*	*78.80*	*77.90*	*74.35*	*77.24*	*73.55*	*71.52*	*81.08*	*81.11*

Figura 5.2: Divergência de Kullback–Leibler entre os domínios presentes na base de dados quando avaliado sobre as 2000 palavras mais frequentes de cada.

DIAGONAL	avaliação \ treino	PERSON	HIST.	CNT.	CITY	ENT.	BIO.	LOC.	TECH.	LAW	RLG.	SPORTS	ORG.	THING	SCHOOL	STATE	LNG.	SCI.
2.68	PERSON	**2.68**	2.93	3.31	3.69	3.81	4.45	3.63	4.45	4.00	3.59	4.45	4.12	4.53	4.38	4.48	4.51	4.75
2.07	HISTORY	2.28	**2.07**	2.78	3.07	3.69	3.93	2.71	4.22	3.38	3.78	3.93	3.50	3.92	4.01	3.54	4.09	4.44
-	COUNTRY	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.27	CITY	3.00	3.10	3.10	**2.27**	3.41	4.31	2.54	3.79	3.95	4.40	3.46	4.15	4.13	3.65	3.13	4.66	4.58
2.39	ENTERTAINMENT	2.70	3.97	3.55	3.76	**2.39**	4.16	3.66	3.88	3.80	4.34	3.59	4.01	4.89	3.99	4.80	5.12	4.99
2.61	BIOLOGY	4.62	4.88	5.00	4.99	4.73	**2.61**	4.90	4.71	4.78	5.41	5.51	5.15	4.61	5.53	5.45	5.17	4.48
2.57	LOCATION	2.93	2.84	2.76	**2.53**	3.46	3.62	2.57	3.94	3.77	4.18	4.02	3.97	3.77	3.77	3.58	4.24	3.97
3.12	TECHNOLOGY	3.52	3.73	4.07	3.75	**2.99**	3.99	4.24	3.12	4.16	5.10	4.19	4.51	4.36	4.56	4.95	5.06	4.35
2.02	LAW	2.12	2.05	2.04	2.68	2.98	2.99	2.48	3.07	**2.02**	3.03	3.26	2.59	3.57	3.16	3.54	3.72	3.70
2.90	RELIGION	**2.43**	2.91	2.79	3.46	3.90	4.15	3.21	4.75	3.11	2.90	4.58	3.75	4.37	3.65	4.14	4.47	4.43
3.20	SPORTS	3.15	3.95	3.81	**3.14**	3.73	5.41	3.91	4.21	4.78	5.52	3.20	4.90	4.83	3.94	4.22	5.60	5.54
4.42	ORGANIZATION	3.42	3.39	3.55	**3.32**	4.49	5.28	3.41	5.53	4.45	3.83	4.94	4.42	4.94	4.70	4.31	4.71	5.56
4.41	THING	3.72	3.56	3.50	3.75	4.00	3.95	3.98	**3.35**	3.65	5.09	4.70	3.91	4.41	4.25	4.62	5.24	4.11
2.12	SCHOOL	2.35	2.74	2.71	2.78	3.10	3.35	2.91	3.67	2.95	3.10	3.62	3.24	3.95	**2.12**	3.41	3.87	4.03
3.52	STATE	2.65	2.92	**2.42**	2.73	3.93	4.11	2.54	4.07	3.24	4.09	3.82	3.85	4.17	3.76	3.52	4.64	4.55
-	LANGUAGE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.89	SCIENCE	2.94	**2.70**	3.48	3.99	3.57	2.97	3.69	3.06	3.68	4.21	4.23	4.06	3.67	4.25	4.56	4.56	2.89

Figura 5.3: Divergência de Kullback–Leibler entre os domínios presentes na base de avaliação e treino.

5.2 Experimentos preliminares

Os primeiros experimentos realizados são dedicados a responder à pergunta QP2. Além da CNN padrão, diferentes redes da literatura foram avaliadas na Figura 5.4. Em particular a ResNet, proposta como uma extensão da CNN e sendo caracterizada por ser uma rede mais profunda e de camadas menores. Também avaliou-se o desempenho de uma rede recorrente simples e uma biLSTM, verificando que ambas têm um desempenho razoável, mas com a LSTM sendo a superior entre as duas. Avaliou-se também o desempenho do modelo híbrido CNN−biLSTM.

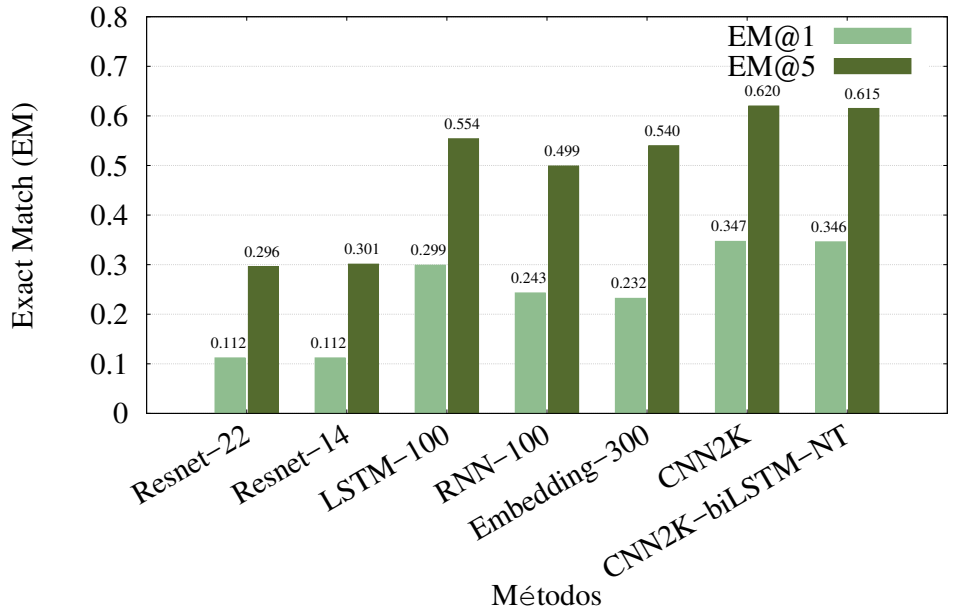


Figura 5.4: Desempenho das diferentes redes implementadas. Avaliadas a Resnet com 22 e 14 camadas convolucionais, uma rede recorrente e uma LSTM com 100 neurônios, uma rede contendo apenas uma camada de embedding conectada a um neurônio, uma rede convolucional com 2000 filtros e o modelo proposto.

Explorou-se, então, a capacidade da CNN como modelo individual (Figura 5.5). No cenário de adaptação de domínio, verificou-se o desempenho de redes especializadas. Podemos observar que alguns domínios são naturalmente mais difíceis que outros: o desempenho do modelo no domínio de ciência é consideravelmente inferior ao desempenho em cidades. Além disso, eles estão distantes da média do modelo no caso geral. Em poucos casos temos um deterioramento do desempenho em função da transferência de aprendizado e, no caso médio, temos um aumento no desempenho do sistema proposto como ilustrado no gráfico onde os modelos sem transferência tem um desempenho inferior. Porém, devido à simplicidade dessa arquitetura, temos poucas opções de metodologias para a adaptação de domínio.

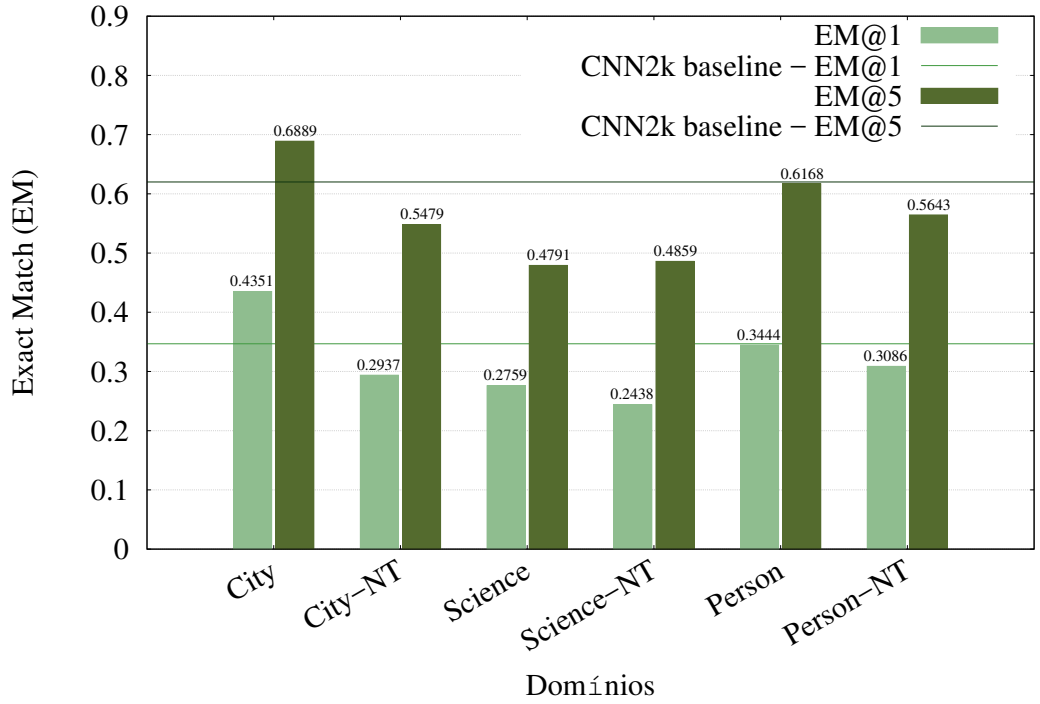


Figura 5.5: Desempenho de CNNs treinadas em domínios específicos. As colunas *Person−NT*, *City−NT* e *Science−NT* representam uma rede treinada apenas no domínio específico, mas avaliada em toda a base de dados. As linhas de *baseline* representam a CNN padrão com 2000 filtros treinada em toda a base de dados e avaliada em todos os dados.

A primeira impressão é que o modelo proposto tem um desempenho inferior à CNN individual. Explorando diferentes configurações de hiper-parâmetros averiguou-se que o desempenho de CNNs com camadas maiores é superior até certo ponto como ilustrado pela Figura 5.6 e em concordância com o trabalho de Feng [Feng et al., 2015a]. O modelo proposto porém, se beneficia de uma camada convolucional ligeiramente menor. Conclui-se que o modelo com apenas uma camada convolucional se beneficia

de mais filtros pois são necessários para cobrir uma maior gama de padrões. O modelo proposto, por outro lado, possui uma camada recorrente e, portanto, é capaz de gerar uma abstração mais complexa, não necessitando dessa quantidade extra de filtros. Na verdade, o aumento indevido da camada convolucional acaba prejudicando o modelo tornando-o mais propenso a sobreajustes e fazendo com que o seu desempenho seja inferior ao da CNN padrão. Também temos o empecilho de que camadas maiores aumentam drasticamente o tempo necessário para a etapa de treino e de adaptação de domínio.

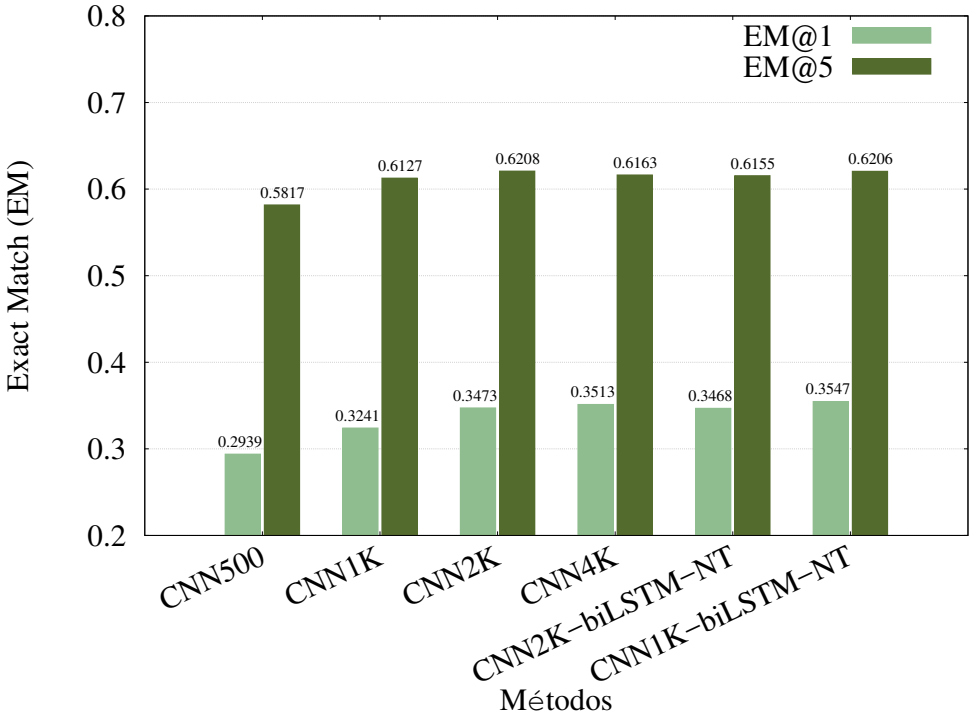


Figura 5.6: Desempenho da CNN e da CNN−biLSTM com diferentes tamanhos de camadas convolucionais. Enquanto a CNN padrão se beneficia de camadas convolucionais maiores, isto causa sobreajuste no modelo proposto.

Como é utilizada uma camada convolucional grande, não é possível empregar uma arquitetura muito profunda uma vez que a carga computacional da etapa de treino se tornaria muito pesada. Uma alternativa é empregar filtros de diferentes tamanhos na mesma camada convolucional. Isso permite buscar padrões em diferentes campos receptivos, sem aumentar o número de hiperparâmetros a serem otimizados. Como todos os filtros estão na mesma camada, é possível alimentar a LSTM com todos os padrões encontrados, independentemente do tamanho. Usando uma implementação padrão, deveriam ser criados atalhos entre as camadas mais rasas e a recorrente. A Figura 5.7 ilustra este experimento. A modelagem com filtros de diferentes ta-

manhos supera quase todos as demais, sendo apenas ligeiramente inferior à de um único filtro de tamanho médio. Embora empregar filtros de tamanhos variáveis não tenha melhorado o desempenho, isso permite aumentar o poder de abstração da rede. Por simplicidade, quando nos referirmos ao modelo CNN−biLSTM (e suas variantes CNN−biLSTM−NT, CNN−biLSTM−DS e CNN−biLSTM−DA) nas próximas seções estamos utilizando a arquitetura com uma camada convolucional menor (1000 filtros). Note, porém, que as conclusões se aplicam a ambas, apenas temos que os valores obtidos são ligeiramente diferentes.

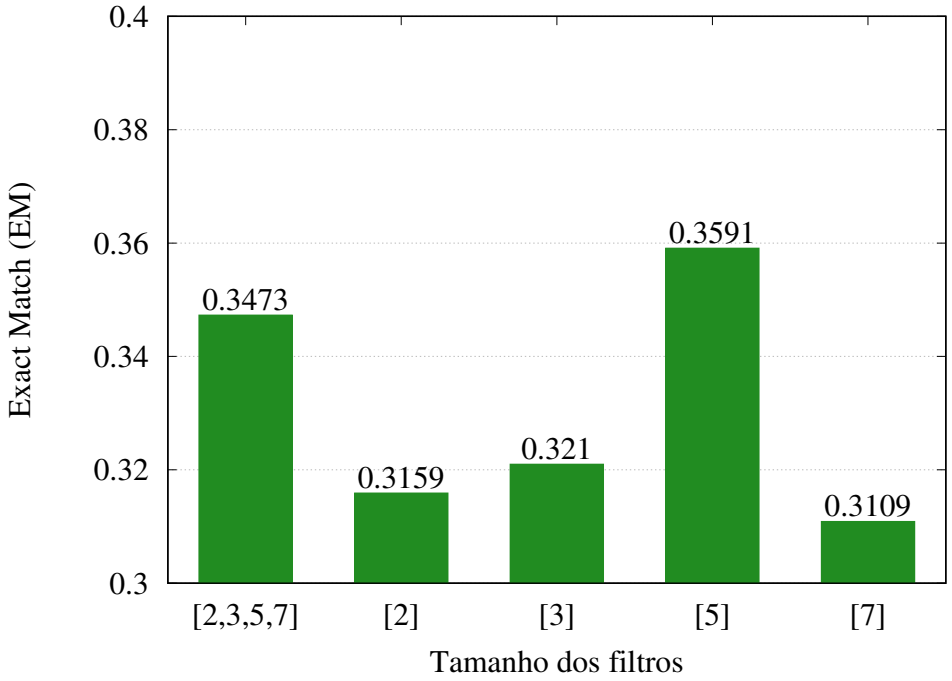


Figura 5.7: Efeitos de empregar diferentes tamanhos de filtros em um modelo convolucional simples. Embora uma das arquiteturas com um único filtro supere a abordagem proposta, ao realizar a transferência de aprendizagem e adicionar mais camadas, a arquitetura com múltiplos tamanhos trará mais beneficios.

5.3 Adaptação de domínio

O primeiro experimento desta seção é dedicado a responder QP3. A Figura 5.8 mostra o desempenho dos dois modelos CNN−biLSTM nos três menores e maiores domínios. Temos que aqueles treinados usando apenas os dados do domínio alvo (CNN−biLSTM−DS) obtiveram os menores valores de EM. Isso pode ser atribuído à pequena quantidade de amostras, o que leva a sobreajustes e à pequena capacidade de generalização. Em contrapartida, o modelo que foi treinado em toda a rede mas

que não foi especializado (CNN−biLSTM−NT) tem um desempenho superior e mais consistente. Observa-se ainda que ao realizarmos adaptação de domínio temos um ganho no desempenho em quase todos os casos, levando a um aumento de sua acurácia no caso médio.

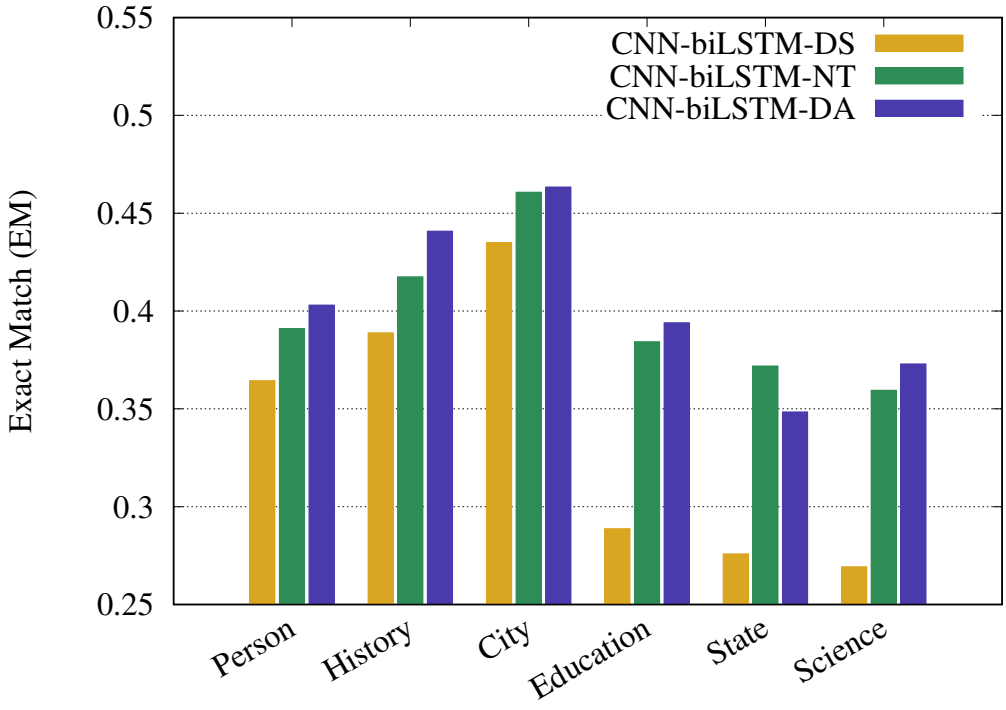


Figura 5.8: Desempenho dos modelos CNN−biLSTM nos três maiores e menores domínios respectivamente. A adaptação de domínio é benéfica em quase todos os casos. Os modelos treinados unicamente no domínio alvo são sempre inferiores.

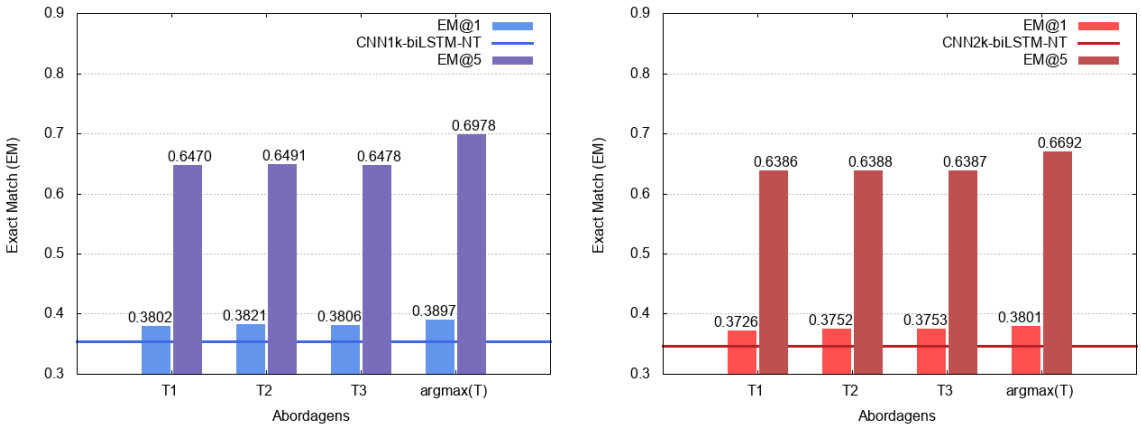
Observando o desempenho de cada um dos três métodos de transferência de aprendizado, vemos que todos são superiores ao modelo não especializado CNN−biLSTM−NT como ilustrado na Tabela 5.1, mas não podemos concluir que são estatisticamente diferentes entre si. Cada método parece superar os demais em um subconjunto de domínios se equiparando quando calculado a acurácia média. Mas, ainda assim, a diferença entre os modelos é pequena. Não podemos dizer que qualquer abordagem de transferência de aprendizado seja estaticamente melhor do que as outras, mas podemos empregar uma estratégia semelhante a uma árvore de decisão para escolhermos sempre o método mais adequado para cada situação, obtendo um ganho de desempenho ainda maior. Note porém que, na maioria dos casos, o modelo treinado no mesmo domínio que o alvo não é aquele com a maior precisão no conjunto de avaliação. Isso parece implicar que as abordagens de divisão de domínio realizadas não são as mais adequadas nesse cenário.

	T1		T2		T3	
	D. Alvo	Melhor D.	D. Alvo	Melhor D.	D. Alvo	Melhor D.
Person	0.401	**0.405**	0.403	0.404	0.401	0.402
History	0.422	0.439	0.440	0.449	0.434	**0.456**
Country	-	-	-	-	-	-
City	0.459	0.459	0.449	0.469	**0.463**	**0.463**
Enter.	0.360	**0.381**	0.361	0.361	0.373	0.373
Biology	**0.296**	**0.296**	0.267	0.274	0.279	0.279
Location	0.372	0.377	0.381	**0.391**	0.383	0.387
Tech	0.326	0.336	0.341	**0.344**	0.334	0.334
Law	0.317	0.320	**0.323**	**0.323**	0.317	0.317
Religion	0.365	0.365	0.344	**0.368**	0.364	0.364
Sports	0.389	0.389	0.388	0.388	0.380	**0.392**
Org.	0.486	0.554	0.527	**0.528**	0.466	0.522
Thing	0.285	0.346	0.326	**0.374**	0.302	0.329
School	0.378	0.378	0.394	0.394	**0.399**	**0.399**
State	0.354	0.398	0.348	**0.400**	0.370	0.388
Language	-	-	-	-	-	-
Science	0.354	0.372	**0.373**	**0.373**	0.367	0.369

Tabela 5.1: Acurácia de cada método de transferência de aprendizado em cada domínio na arquitetura CNN−biLSTM−DA no problema de *spans*. Valores na coluna "D. Alvo" são aqueles onde o modelo foi treinado no mesmo domínio que o alvo. A coluna "Melhor D." representa as melhores pontuações de EM (*Exact Match*) obtidas em cada abordagem de transferência de aprendizado, independentemente em onde o modelo foi treinado. Valores destacados estão associados aos maiores valores de EM em cada cenário.

5.4 Pergunta−resposta sobre sentenças

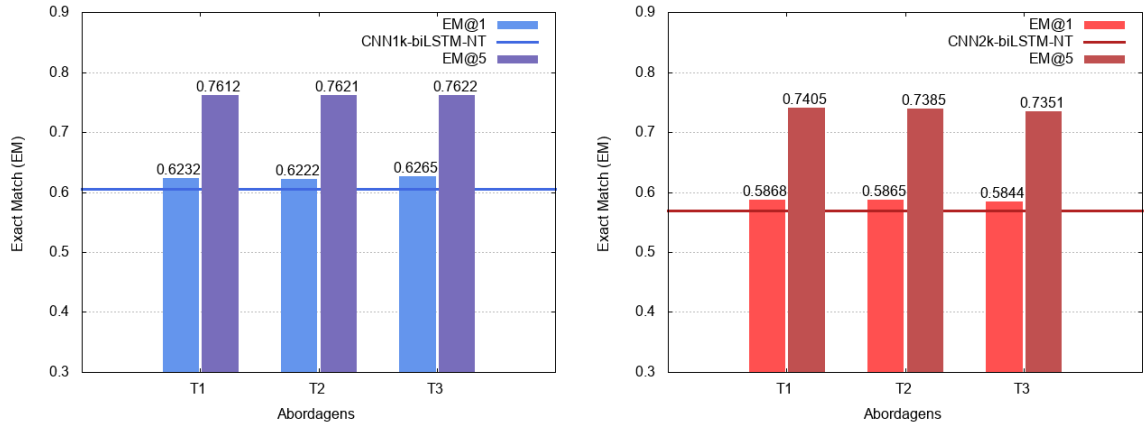
Um segundo conjunto de experimentos concentra-se no problema de QAs focado no nível de sentenças. Ao invés de tentar prever o conjunto correto de *spans*, deseja-se prever a sentença que contém a resposta correta. Esse problema é ligeiramente mais fácil do que o anterior, pois as redes recebem uma quantidade maior de informação e temos um compartilhamento maior de n−gramas entre pergunta e resposta e as figuras 5.9 e 5.10 ilustram os resultados dos experimentos no problema de *spans* e sentenças respectivamente. Como esperado, o modelo proposto atinge uma acurácia na classificação muito maior. Realizar a adaptação do domínio, da mesma forma, exerce melhorias semelhantes ao problema envolvendo *spans*. À primeira vista, ao analisar o desempenho de cada modelo treinado em cada domínio, chega-se à conclusão de que o domínio alvo pode não ser a melhor base de treino.



(a) Desempenho da CNN−biLSTM−DA com uma menor camada convolucional em nível de *spans*.

(b) Desempenho da CNN−biLSTM−DA com uma maior camada convolucional em nível de *spans*.

Figura 5.9: Acurácia de cada abordagem de transferência de aprendizado quando comparada com nenhum *finetunning*. Resultados relativos às arquiteturas com diferentes tamanhos de camadas convolucionais.



(a) Desempenho da CNN−biLSTM−DA com uma menor camada convolucional em nível de sentença.

(b) Desempenho da CNN−biLSTM−DA com uma maior camada convolucional em nível de sentença.

Figura 5.10: Acurácia de cada abordagem de transferência de aprendizado quando comparada com nenhum *finetunning* no problema das sentenças. Resultados relativos às arquiteturas com diferentes tamanhos de camadas convolucionais.

É apresentada uma comparação entre diferentes modelos CNN−biLSTM−DA aprendidos seguindo as três abordagens propostas de transferência de aprendizado tentando ainda responder a QP4. Porém, os experimentos foram ampliados para conterem os resultados tanto em nível de *spans* quanto em nível de sentenças. A Tabela 5.2 mostra números de EM para cada domínio. Os números de EM variam bastante de-

pendendo do domínio, assim como a melhor abordagem de transferência. Condizendo com os resultados obtidos no experimento anterior, algumas abordagens se destacam em certos cenários e que cada modelo CNN−biLSTM−DA supera os demais em um subconjunto de domínios.

Ainda em relação ao QP4, a Tabela 5.3 mostra os valores globais para EM nos QAs em nível de *spans* e de sentenças. Nesse caso, o desempenho é avaliado considerando-se todo o conjunto de questões. Novamente, a adaptação do domínio é sempre benéfica para o desempenho final dos modelos CNN−biLSTM−DA, já que todos os três modelos são superiores aos CNN−biLSTM−DS e CNN−biLSTM−NT.

A última coluna da Tabela 5.2 mostra o desempenho do QA no nível de *spans* alcançado quando exploramos adicionar evidências retiradas das sentenças e é dedicada a responder QP5. Vemos que o desempenho do QA de *spans* é amplamente impulsionado quando empregamos as informações do nível de sentença. Um exemplo interessante nos ajudará a entender os motivos que levaram a essa melhoria. Considere a questão *'Qual equipe ganhou a última Liga dos campeões da UEFA?'*. Ambos *'Barcelona'* e *'Real Madrid'* são respostas candidatas adequadas. Essas respostas foram extraídas das respectivas frases:

- *"A vitória na final da Liga dos Campeões resultou no* ***Real Madrid*** *ser o primeiro time a defender com sucesso seu título na era da UEFA Champions League."*
- *"****Barcelona*** *é um dos principais centros turísticos, econômicos, comerciais e culturais do mundo."*

Ao inspecionar as duas frases, torna-se trivial concluirmos qual a resposta correta. No entanto, não temos acesso a esse tipo de informação no nível de *spans*. De maneira geral, utilizar as evidências do nível de sentença leva a uma melhoria de 39% no desempenho em nível de *spans*. Como mostrado na Figura 5.11, quanto maior o desempenho, maior será o ganho quando fornecida esse tipo de informação.

Embora esses resultados sejam interessantes por si só, a maior descoberta vem de procurar as melhores combinações de modelos de nível *span* e nível de sentença para cada domínio. Por meio de força bruta, avaliamos todas as combinações de modelos utilizando as três abordagens de transferência de aprendizado. Raramente temos que o modelo de *spans* escolhido foi treinado no domínio alvo, ao contrário do envolvendo sentenças, onde frequentemente o modelo mais apropriado foi treinado no mesmo domínio que o alvo. Ao inspecionar a relação entre as bases de treino e avaliação, descobrimos também que os temas *Sports*, *Organization* e *Thing* são aqueles

Tabela 5.2: Valores de EM em nível de *span* e nível de sentença para diferentes abordagens de transferência de aprendizado em cada domínio. A última coluna mostra o EM obtido combinando ambos modelos. Os melhores resultados de cada linha estão destacados e não possuem uma diferença estatiscamente significativa.

	Nível de *span* (f^d_{span})			Nível de sentença (f^d_{sent})			
Domínio	T1	T2	T3	T1	T2	T3	$f^d_{span} + f^d_{sent}$
Person	**0.405**	**0.404**	**0.402**	**0.561**	**0.565**	**0.561**	0.504
History	0.439	0.449	**0.456**	**0.625**	**0.624**	0.619	0.590
City	0.459	**0.469**	0.463	**0.683**	0.672	0.670	0.656
Entertainment	**0.381**	0.361	0.373	0.578	**0.584**	0.572	0.519
Biology	**0.296**	0.274	0.279	**0.591**	0.574	0.576	0.444
Location	0.377	**0.391**	**0.387**	0.644	**0.648**	0.640	0.513
Technology	0.336	**0.344**	0.334	0.630	0.627	**0.641**	0.483
Law	0.320	**0.323**	0.317	0.604	0.603	**0.611**	0.470
Religion	**0.365**	**0.368**	**0.364**	**0.691**	**0.688**	0.611	0.542
Sports	**0.389**	0.388	**0.392**	0.633	0.639	**0.702**	0.492
Organization	**0.554**	0.528	0.522	0.628	0.642	**0.662**	0.671
Thing	0.346	**0.374**	0.329	**0.772**	**0.775**	0.768	0.609
Education	0.378	0.394	**0.399**	**0.645**	0.638	**0.646**	0.522
State	**0.398**	**0.400**	0.388	0.707	**0.720**	0.714	0.597
Science	**0.372**	**0.373**	0.369	0.625	0.621	**0.638**	0.503
Mínimo	0.296	0.274	0.279	0.561	0.565	0.561	0.444
Máximo	0.554	0.528	0.522	0.772	0.775	0.768	0.671

Tabela 5.3: Desempenho geral dos modelos CNN−biLSTM considerando o *Exact Matching*. Melhores resultados encontram-se destacados.

			CNN−biLSTM−DA		
	CNN−biLSTM−DS	CNN−biLSTM−NT	T1	T2	T3
Spans	0.311	0.352	**0.380**	**0.382**	**0.381**
Sentenças	0.542	0.601	**0.633**	**0.622**	**0.626**

com a maior divergência de KL entre as bases. Considerando isso, é adequado dizer que a divisão de domínios por temas e o QA envolvendo sentenças têm alguma correlação.

Se imaginarmos o cenário onde a resposta correta é a segunda classificada no QA envolvendo *spans* e a frase de onde ela foi retirada também é a segunda classificada no QA de sentenças, nenhum dos modelos é capaz de acertar essa pergunta. Consideremos

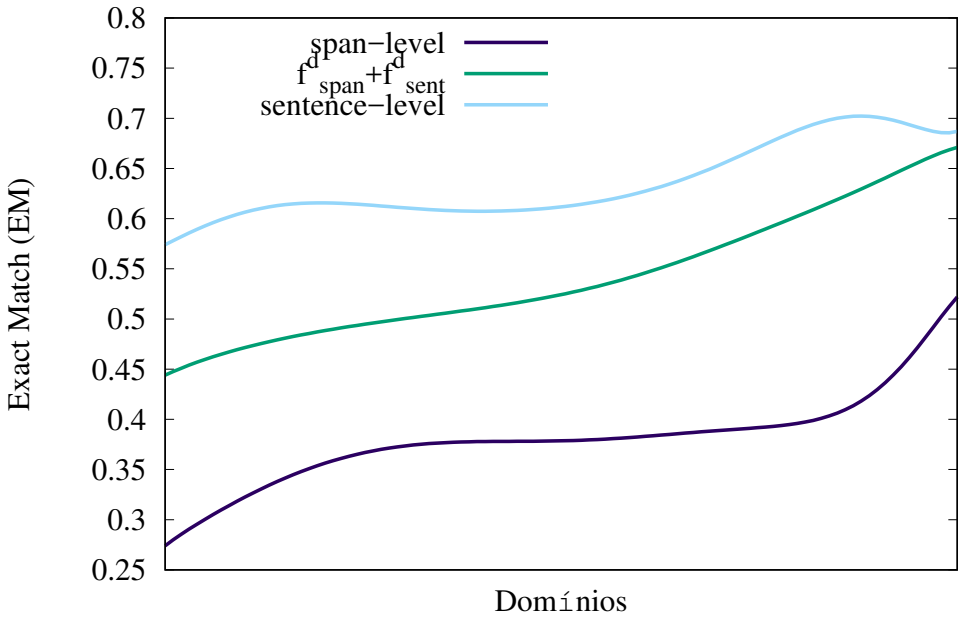


Figura 5.11: Domínios estão ordenados em função da acurácia ao nível de *spans*. Quanto melhor o desempenho, maior o ganho ao combinar o resultado do QA de *spans* com o QA de sentenças

aplicar $f^d_{span} + f^d_{sent}$, isto é, combinar as saídas dos modelos em nível de *span* e de sentenças. Caso a frase do *span* mais relevante não possuir uma classificação igualmente alta no QA de sentenças, sua relevância combinada não será alta. Temos uma situação análoga quando os *spans* de uma frase relevante não têm uma boa classificação. No cenário hipotético criado temos uma relevância combinada alta visto que as alternativas corretas eram a segunda classificada em ambos os modelos. Alguns exemplos dessa situação ocorrem no domínio *Organization*. A acurácia combinada dos modelos é superior à do QA envolvendo sentenças. Seguem as perguntas que o modelo combinado acerta mas que nenhum dos QAs individuais é capaz de obter a resposta correta:

- *'When did this attempt take place?' ('1560')*
- *'From whom did the Huguenots in South Carolina purchase land from?' ('Edmund Bellinger')*
- *'What Irish cities had Huguenot mayors in the 1600s and 1700s?' ('Dublin, Cork, Youghal and Waterford')*
- *'What was the name of France's primary colony in the New World?' ('New France')*

- *'What did the Edict do for Huguenots in France?' ('granted the Protestants equality with Catholics under the throne and a degree of religious and political freedom within their domains')*
- *'From what French King did the Huguenot name possibly descend?' ('Hugues Capet')*
- *'From whom did the Huguenots purchase the land where they settled?' ('John Pell, Lord of Pelham Manor')*

Também foi verificado o desempenho comparativo dos métodos CNN−biLSTM−DS e CNN−biLSTM−NT à adaptação de domínio ao nível de sentenças. Similar ao experimento envolvendo *spans*, percebemos que domínios maiores não são tão afetados ao treinar os modelos exclusivamente no alvo porém em domínios menores, o desempenho do modelo treinado em toda a base sobressai ao específico. Todavia, em todos os casos, a adaptação de domínio mostrou ser a abordagem superior como ilustrado pela Figura 5.12.

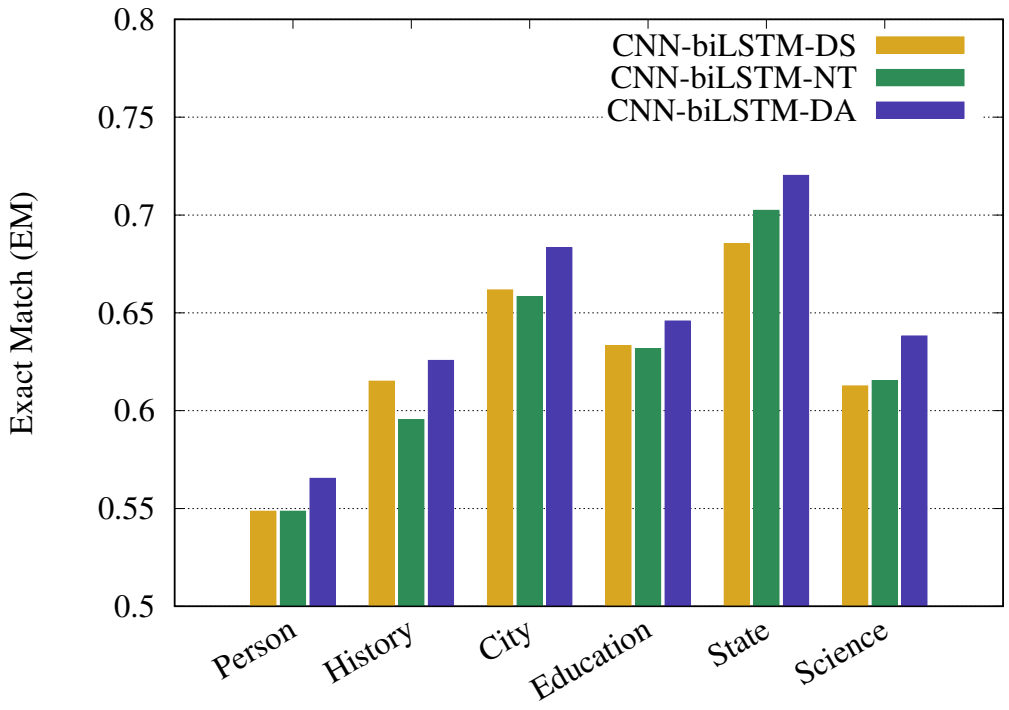


Figura 5.12: Desempenho dos modelos CNN−biLSTM nos menores e maiores domínios ao nível de sentenças. A adaptação do domínio é benéfica em todos os casos. Os modelos treinados unicamente no domínio alvo são inferiores nos menores domínios e conseguem ser superiores ao modelo treinado em todos os dados nos maiores domínios. Isso pode ser atribuído ao fato do problema em nível de sentenças ser mais fácil.

5.5 Divisão automática de domínios

O próximo conjunto de experimentos é dedicado a responder QP6. Até agora, assumimos que os tópicos são dados a priori, de modo que os domínios e tópicos são bem definidos. Investigamos como o desempenho de QAs é afetado pela aplicação de métodos simples de identificação de temas. Para isso, treinou-se uma representação usando Doc2Vec sobre os parágrafos da base de treino e, em seguida, executou-se o K−Means para tentar dividir os dados em domínios. Foram explorados um total de seis abordagens diferentes: três representações de Doc2Vec aplicados na divisão em cinco e dezessete *clusters*. A Tabela 5.4 resume o desempenho do modelo usando a melhor abordagem de divisão automática de domínios (E3).

Nos experimentos envolvendo cinco domínios temos que em todos os casos o modelo de sentença selecionado foi treinado no domínio alvo. Mesmo no experimento com dezessete *clusters*, onde os dados estão mais estratificados, ainda temos que, em 85% dos cenários avaliados, o modelo de sentença mais interessante foi treinado no domínio alvo. Como o QA envolvendo sentenças tem uma sobreposição de n−gramas com o texto, esse resultado já era esperado como relatado na seção anterior. Como a divisão dos domínios é feita utilizando os *embeddings* dos parágrafos e, por consequência seu vocabulário, essa interação acabou sendo potencializada nesses experimentos. Esse resultado confirma nossa hipótese inicial de que a especialização de modelos de QA em função dos temas abordados é uma estratégia significativa. Os resultados encontram-se sumarizados nas figuras 5.13 e 5.14.

Finalmente, foi realizada uma análise comparativa visando responder QP7. Para isso avaliamos o desempenho do QA em nível de *spans* contra quatro modelos recentemente propostos e que utilizam a base do SQuAD. A Figura 5.15 indica o desempenho em comparação com os *baselines* descritos. Técnicas diversas são empregadas utilizando-se diferentes algoritmos. Ressalta-se que um deles chega a inclusive criar dados sintéticos a fim de enriquecer sua base de treino. Ainda assim, os modelos CNN−biLSTM−DA foram capazes de alcançar um EM superior. O melhor resultado obtido foi o do modelo CNN1k−biLSTM−DA que utiliza as melhores abordagens de transferência de aprendizado para cada caso e emprega uma divisão automática de domínios por meio da estratégia E3 e usando dezessete *clusters*.

Domínio alvo	f^d_{span}	f^d_{sent}	EM
index0	**T3-index0**	**T1-index0**	0.5254
index1	T3-index2	**T2-index1**	0.4785
index2	**T3-index2**	**T1-index2**	0.5589
index3	**T2-index3**	**T2-index3**	0.4300
index4	T3-index1	**T1-index4**	0.5536
Total	*0.3941*	*0.6258*	*0.5082*

Domínio alvo	f^d_{span}	f^d_{sent}	EM
index0	T1-index15	**T2-index0**	0.4185
index1	T3-index2	T3-index5	0.5700
index2	T2-index15	**T1-index2**	0.5100
index3	T1-index14	**T1-index3**	0.5906
index4	**T3-index4**	**T1-index4**	0.6104
index5	T3-index1	**T1-index5**	0.5030
index6	**T3-index6**	**T1-index6**	0.5382
index7	T3-index1	**T2-index7**	0.4717
index8	**T3-index8**	**T2-index8**	0.5209
index9	T1-index12	**T2-index9**	0.5171
index10	T3-index3	**T1-index10**	0.5421
index11	T3-index6	T1-index15	0.5924
index12	T3-index1	**T2-index12**	0.6060
index13	T3-index0	**T3-index13**	0.4639
index14	T1-index6	T3-index5	0.6137
index15	T1-index3	**T2-index15**	0.4848
index16	T2-index2	**T1-index16**	0.5126
Total	*0.3997*	*0.6432*	*0.5221*

Tabela 5.4: Acurácia das melhores combinações de modelos no nível de *spans* f^d_{span} e sentença f^d_{sent} usando a representação E3 para cinco e dezessete *clusters* respectivamente. As células ilustram qual a melhor combinação de abordagem de transferência de aprendizado e domínio para cada modelo de *spans* e sentenças. Enquanto a escolha do melhor modelo em nível de spans não seja uma tarefa trivial, é possível observar que em quase todos os casos o modelo em nível de sentença selecionado foi treinado no mesmo domínio que o alvo, como ilustrado pela células em destaque.

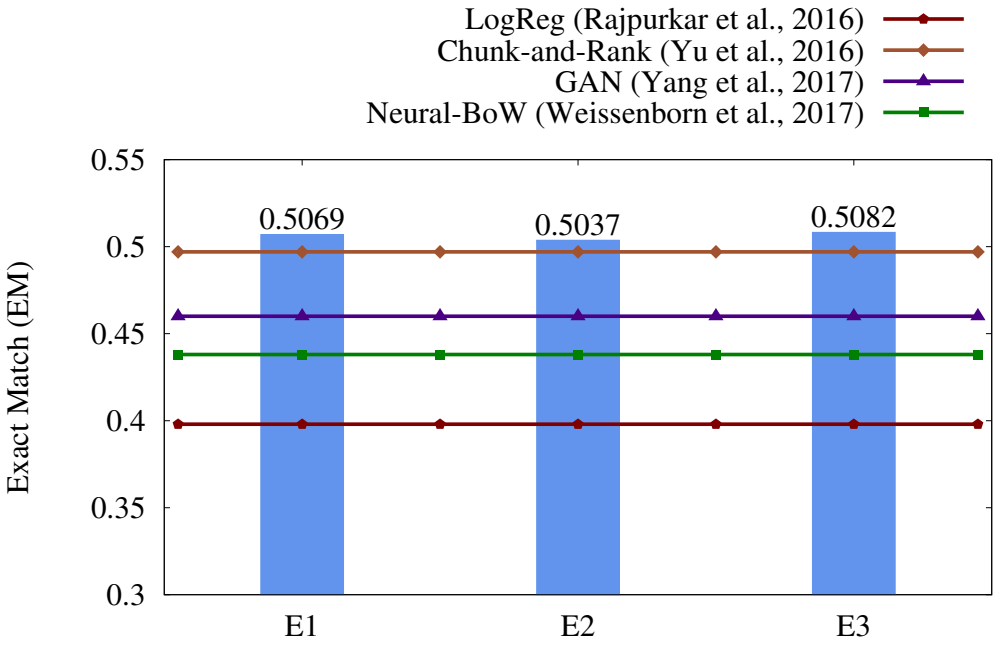


Figura 5.13: Desempenho em nível de *spans* obtido pelo modelo CNN−biLSTM−DA assumindo as três estratégias propostas para divisão automática de cinco domínios comparados ao desempenho de *baselines* recentes. A abordagem onde foi treinada uma nova representação de Doc2Vec com uma janela de 15 palavras é a superior (E3).

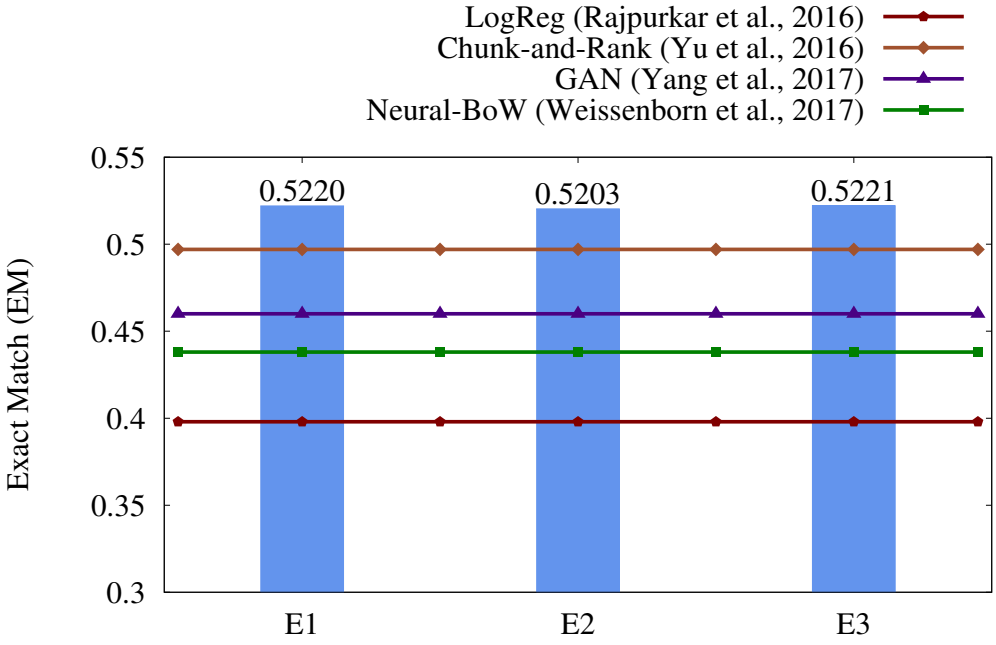


Figura 5.14: Desempenho em nível de *spans* obtido pelo modelo CNN−biLSTM−DA assumindo as três estratégias propostas para divisão automática de dezessete domínios comparados ao desempenho de *baselines* recentes. A abordagem onde foi treinada uma nova representação de Doc2Vec com uma janela de 15 palavras é a superior (E3).

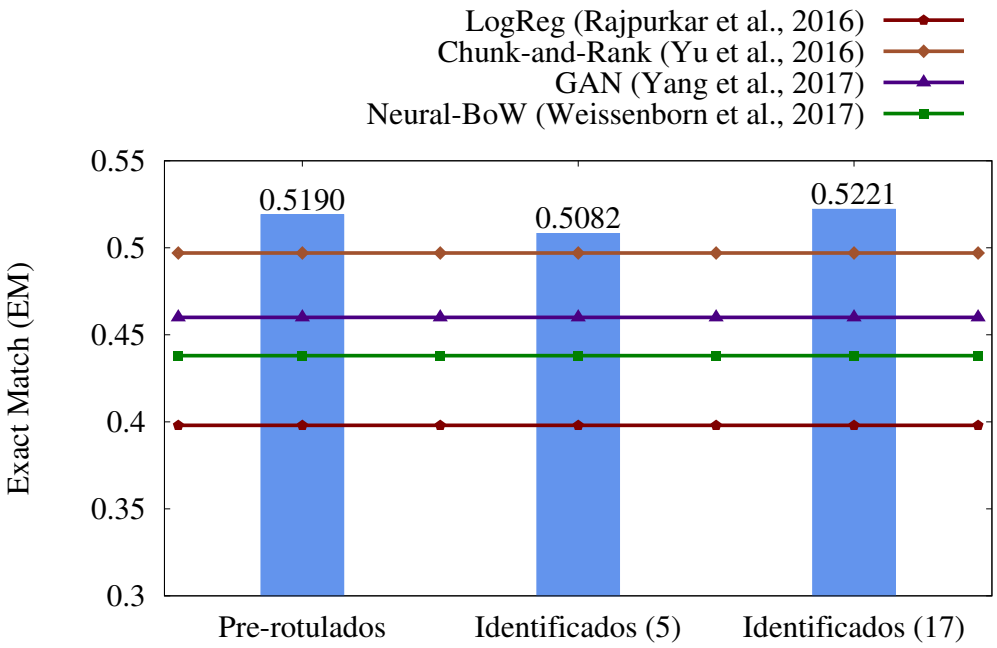


Figura 5.15: Desempenho em nível de *spans* obtido pelo modelo CNN−biLSTM−DA assumindo três cenários. À esquerda, os domínios dos tópicos são explicitamente dados. Ao centro, os domínios dos tópicos são identificados por um método de *clusterização* simples. À direita, empregamos o mesmo método de *clusterização*, mas utilizando o mesmo número de domínios adotados no cenário onde eles são explicitamente dados. A figura também mostra o desempenho de *baselines* recentes. Todos os métodos propostos são capazes de bater os baselines apresentados, sendo o modelo utilizando a divisão automática para dezessete domínios o superior.

Capítulo 6

Conclusões e trabalhos futuros

Neste trabalho foi proposto aplicar uma estratégia semelhante à divisão e conquista para o problema de QAs multi-domínio. Dividiu-se nossa base em vários subconjuntos e treinamos um modelo especializado em cada. Foi proposta a adaptação do domínio com base no tema implícito das perguntas e o assunto do artigo de onde elas foram retiradas, avaliando-se diferentes abordagens de transferência de aprendizado. Concluímos que nenhuma é superior às demais no caso médio, mas elas se sobressaem umas as outras em casos específicos. Podemos aproveitar dessa característica e escolher sempre a melhor abordagem para cada domínio, impulsionando ainda mais a acurácia do sistema. Os experimentos também demonstraram que a adaptação de domínio usando informações implícitas de contexto também geram um aumento de desempenho.

Em nível de *spans*, nem sempre é trivial a escolha do melhor modelo especializado. Em muitos casos, a rede especializada no domínio alvo não é aquela de melhor desempenho. Isso nos leva a crer que o tema de cada parágrafo talvez não seja a melhor opção para esse tipo de problema. Note, porém, que os resultados dos modelos especializados no domínio alvo são próximos dos ótimos. Uma heurística razoável poderia ser sempre utilizar o modelo treinado no domínio alvo. Isso não ocorre no nível de sentença, que é mais previsível. A adaptação diretamente no domínio alvo é altamente eficiente, proporcionando alguns dos melhores resultados.

Consideremos duas perguntas relacionadas à cidade de Brasília, "*Quando Brasília foi inaugurada?*" e "*Quem inaugurou Brasília?*". Ambas as perguntas estão associadas ao tema de uma cidade, mas suas respostas, '*1960*' e '*Juscelino Kubitschek*', são extremamente diferentes. Aprender a associar essas respostas a '*Brasília*' ajuda na resposta da pergunta, mas a associação entre '*1960*' e '*Juscelino Kubitschek*' não necessariamente traz ganhos consideráveis. Essas duas respostas foram retiradas da frase '*Inaugurada em 21 de abril de 1960, pelo então presidente Juscelino Kubitschek, Brasí-*

lia tornou-se formalmente a terceira capital do Brasil, após Salvador e Rio de Janeiro.'. Ambas as perguntas estão associadas a uma mesma resposta no QA em nível de sentença. Se a pergunta se refere a uma data, uma pessoa ou até mesmo um local, não é relevante. Nesse nível mais alto, estamos preocupados unicamente com o contexto da pergunta. Isso nos ajuda a explicar seu desempenho mais previsível durante a adaptação de domínio.

Outra contribuição é uma maneira simples e rápida de condicionar a escolha da resposta correta levando em consideração frases relevantes para a questão. Em QA ao nível de *spans*, as respostas são apenas algumas palavras com quase nenhuma conexão com o restante do contexto dos parágrafos. Por outro lado, nas respostas do QA em nível de sentenças, estas são subdivisões maiores do parágrafo e, portanto, uma correlação maior pode ser explorada. Para isso, aprendemos de forma independente, modelos para QAs tanto de sentença quanto de *spans*. Então, condicionamos a escolha da resposta como um problema de busca usando os dois modelos em tempo de teste.

Mostrou-se também os benefícios da divisão automática de domínios. Foi empregada uma técnica relativamente simples e que utiliza algoritmos consolidados na literatura. Para o mesmo número de domínios de nossa rotulação manual, o método automático proporcionou modelos de QA superiores em todos os casos. O método com apenas 5 *clusters* teve um desempenho inferior aos dos demais, mas ele tem a vantagem de ser avaliado muito mais rápido e também é capaz de bater os *baselines* propostos. Independente do número de domínios, na etapa de treino processaremos o mesmo número de perguntas. Porém, na etapa de avaliação, observamos o desempenho de cada modelo em cada domínio. Assim, o tempo de avaliação aumenta na ordem $O(n)$ onde n representa o número de domínios. Porém, devemos testar todas as possibilidades de combinações de modelos de *spans* e sentenças em nossa abordagem hibrida, levando a uma complexidade $O(n^2)$ de tempo. Para 5 domínios, considerando os três métodos de transferência, temos um total de 225 combinações. No cenário com 17 domínios, esse número aumenta para 2601.

De maneira geral, os resultados indicam que a adaptação do domínio é efetiva, levando a ganhos de acurácia que chegam a 20% em alguns domínios. Na média, os modelos têm um aumento de desempenho de 10% ao realizar a adaptação. O condicionamento do modelo de *spans* ao de sentenças também é muito eficaz, já que observou-se um aumento de 40% no seu desempenho.

Explorar outras arquiteturas certamente nos ajudará a criar um modelo mais robusto. A arquitetura empregada é relativamente simples, contendo apenas quatro camadas: uma de *embeddings*, uma convolucional, uma recorrente e uma totalmente conectada. O trabalho foi fortemente inspirado na adaptação de domínio utilizada

nos campos de análise de imagens e áudio. Neles, as redes costumam ser bem mais complexas contendo múltiplas camadas convolucionais e recorrentes.

Uma grande dificuldade da modelagem proposta é a necessidade da extração de *spans* candidatos. Este problema não foi explorado, mas entendemos que ele por si só é um tópico relevante de pesquisa. Muitos trabalhos envolvendo sistemas de pergunta−resposta aproveitam da estrutura de *pointer networks* para lidar com essa tarefa. Para diminuir a carga computacional desse processo, é comum extrair passagens de seções relevantes dos parágrafos usando mecanismos de atenção. Utilizar essas duas técnicas está relacionado com explorar novas arquiteturas.

Finalmente, investigar outras técnicas de transferência de aprendizado e divisão de domínios certamente trará benefícios. Em nossos experimentos observamos que o tema dos parágrafos e o QA ao nível de sentença têm uma forte correlação, ao contrário do QA ao nível de *spans*. Uma hipótese é que as perguntas e respostas desse problema estão fortemente relacionadas ao tipo de resposta esperado. Acreditamos que perguntas relacionadas com datas têm uma interação maior entre si que perguntas de um mesmo tema no âmbito dos *spans* por exemplo. Uma abordagem sugerida seria explorar o tipo esperado de resposta para a divisão de domínios, talvez por meio das *WH-words* (palavras *What, When, Where*, etc.) de cada pergunta. Poderíamos, portanto, combinar o QA em nível de *spans* focado nas características das perguntas com o QA em nível de sentenças focado no seu contexto.

Apêndice A

Lista de Siglas

ANN: Artificial Neural Network

BoW: Bag-of-Words

CNN: Convolutional Neural Network

EM: Exact Match

GAN: Generative Adversarial Network

GloVe: Global Vectors for Word Representation

IR: Information Retrieval

KL: Kullback–Leibler

L-BFGS: Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm

LSTM: Long Short-term Memory Network

MLP: Multi-layer Perceptron

QA: Question Answering

Resnet: Residual Network

RNN: Recorrent Neural Network

SQuAD: Stanford Question Answering Dataset

TREC: Text REtrieval Conference

Referências Bibliográficas

[Ahmed et al., 2008] Ahmed, A.; Yu, K.; Xu, W.; Gong, Y. & Xing, E. (2008). Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. *Computer Vision–ECCV 2008*, pp. 69--82.

[Ahn et al., 2004] Ahn, D.; Jijkoun, V.; Mishne, G.; Müller, K.; de Rijke, M. & Schlobach, S. (2004). Using wikipedia at the trec qa track.

[Arnold et al., 2007] Arnold, A.; Nallapati, R. & Cohen, W. W. (2007). A comparative study of methods for transductive transfer learning. Em *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pp. 77--82. IEEE.

[Bhatia et al., 2016] Bhatia, S.; Lau, J. H. & Baldwin, T. (2016). Automatic labelling of topics with neural embeddings. Em *26th COLING International Conference on Computational Linguistics*, pp. 953--963.

[Bollacker et al., 2008] Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T. & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. Em *In SIGMOD Conference*, pp. 1247--1250.

[Bordes et al., 2014a] Bordes, A.; Chopra, S. & Weston, J. (2014a). Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

[Bordes et al., 2015a] Bordes, A.; Usunier, N.; Chopra, S. & Weston, J. (2015a). Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

[Bordes et al., 2015b] Bordes, A.; Usunier, N.; Chopra, S. & Weston, J. (2015b). Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

[Bordes et al., 2014b] Bordes, A.; Weston, J. & Usunier, N. (2014b). Open question answering with weakly supervised embedding models. Em *Joint European Confe-*

rence on Machine Learning and Knowledge Discovery in Databases, pp. 165--180. Springer.

[Buscaldi & Rosso, 2006] Buscaldi, D. & Rosso, P. (2006). Mining knowledge from wikipedia for the question answering task. Em *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 727--730.

[Callan, 2004] Callan, J. (2004). Lecture in open domain question answering. Carnegie Mellon University.

[Caruana, 1995] Caruana, R. (1995). Learning many related tasks at the same time with backpropagation. Em *Advances in neural information processing systems*, pp. 657--664.

[Caudill, 1989] Caudill, M. (1989). Neural nets primer, part vi. *AI Expert*, 4(2):61--67.

[Chen & Zhang, 2013] Chen, Z. & Zhang, W. (2013). Domain adaptation with topical correspondence learning. Em *23rd IJCAI International Joint Conference on Artificial Intelligence*, pp. 1280--1286.

[Coutinho et al., 2014] Coutinho, E.; Deng, J. & Schuller, B. (2014). Transfer learning emotion manifestation across music and speech. Em *Neural Networks (IJCNN), 2014 International Joint Conference on*, pp. 3592--3598. IEEE.

[Dang et al., 2007] Dang, H. T.; Kelly, D. & Lin, J. J. (2007). Overview of the trec 2007 question answering track. Em *TREC*, volume 7, p. 63.

[Feng et al., 2015a] Feng, M.; Xiang, B.; Glass, M. R.; Wang, L. & Zhou, B. (2015a). Applying deep learning to answer selection: A study and an open task. Em *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pp. 813--820. IEEE.

[Feng et al., 2015b] Feng, M.; Xiang, B.; Glass, M. R.; Wang, L. & Zhou, B. (2015b). Applying deep learning to answer selection: A study and an open task. Em *2015 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 813--820.

[Ferrucci et al., 2010] Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A. A.; Lally, A.; Murdock, J. W.; Nyberg, E.; Prager, J. et al. (2010). Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59--79.

[Forgy, 1965] Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21:768--769.

[Ghung et al., 2004] Ghung, H.; Song, Y.-I.; Han, K.-S.; Yoon, D.-S.; Lee, J.-Y.; Rim, H.-C. & Kim, S.-H. (2004). A practical qa system in restricted domains.

[Green Jr et al., 1961] Green Jr, B. F.; Wolf, A. K.; Chomsky, C. & Laughery, K. (1961). Baseball: an automatic question-answerer. Em *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pp. 219--224. ACM.

[Hanna, 2011] Hanna, J. (2011). Computer finishes off human opponents on 'jeopardy!'. `http://edition.cnn.com/2011/TECH/innovation/02/16/jeopardy.watson/index.html`. Acessado em: 2016-07-14.

[Harabagiu et al., 2000] Harabagiu, S. M.; Paşca, M. A. & Maiorano, S. J. (2000). Experiments with open-domain textual question answering. Em *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pp. 292--298. Association for Computational Linguistics.

[Harvey, 1994] Harvey, R. L. (1994). *Neural network principles*. Prentice-Hall, Inc.

[Hirschman & Gaizauskas, 2001] Hirschman, L. & Gaizauskas, R. (2001). Natural language question answering: the view from here.

[Hochreiter & Schmidhuber, 1997] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735--1780.

[Hu et al., 2014] Hu, B.; Lu, Z.; Li, H. & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. Em *28th NIPS Annual Conference on Neural Information Processing Systems*, pp. 2042--2050.

[Hubel & Wiesel, 1968] Hubel, D. H. & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215--243.

[Jaech et al., 2016] Jaech, A.; Heck, L. & Ostendorf, M. (2016). Domain adaptation of recurrent neural networks for natural language understanding. *arXiv preprint arXiv:1604.00117*.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[Kullback & Leibler, 1951] Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79--86.

[Lau & Baldwin, 2016] Lau, J. H. & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. Em *In Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 78--86. ACL.

[Le & Mikolov, 2014] Le, Q. & Mikolov, T. (2014). Distributed representations of sentences and documents. Em *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188--1196.

[LeCun et al., 1998] LeCun, Y.; Bottou, L.; Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324.

[Lee et al., 2016] Lee, K.; Kwiatkowski, T.; Parikh, A. P. & Das, D. (2016). Learning recurrent span representations for extractive question answering. *CoRR*, abs/1611.01436.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. Em *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281--297. Oakland, CA, USA.

[Manning et al., 2008] Manning, C. D.; Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. ISBN 0521865719, 9780521865715.

[Marczewski et al., 2017] Marczewski, A.; Veloso, A. & Ziviani, N. (2017). Learning transferable features for speech emotion recognition. Em *ACM MultiMedia*.

[Mikolov et al., 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Em *27th Annual Conference on Neural Information Processing Systems*, pp. 3111--3119.

[Moldovan et al., 2003] Moldovan, D.; Paşca, M.; Harabagiu, S. & Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133--154.

[NIDA, 2007] NIDA, N. I. o. D. A. (2007). Brain power: Grades 6-9. `https://www.drugabuse.gov/publications/brain-power/brain-power-grades-6-9`. Acessado em 22 Agosto, 2017.

[Pan & Yang, 2010] Pan, S. J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345--1359.

[Pasca & Harabagiu, 2001] Pasca, M. & Harabagiu, S. (2001). The informative role of wordnet in open-domain question answering. Em *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*, pp. 138--143.

[Pennington et al., 2014] Pennington, J.; Socher, R. & Manning, C. D. (2014). Glove: Global vectors for word representation. Em *EMNLP*, volume 14, pp. 1532--1543.

[Rajpurkar et al., 2016] Rajpurkar, P.; Zhang, J.; Lopyrev, K. & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

[Rosenblatt, 1957] Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

[Sakai, 2014] Sakai, T. (2014). Statistical reform in information retrieval? *SIGIR Forum*, 48(1):3--12.

[Seo et al., 2016] Seo, M. J.; Kembhavi, A.; Farhadi, A. & Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

[Severyn & Moschitti, 2015] Severyn, A. & Moschitti, A. (2015). Learning to rank short text pairs with convolutional deep neural networks. Em *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 373--382. ACM.

[Shin et al., 2016] Shin, H.-C.; Roth, H. R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D. & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285--1298.

[Simmons, 1965] Simmons, R. F. (1965). Answering english questions by computer: a survey. *Communications of the ACM*, 8(1):53--70.

[Steinhaus, 1956] Steinhaus, H. (1956). Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1(804):801.

[Stroh & Mathur, 2016] Stroh, E. & Mathur, P. (2016). Question answering using deep learning.

[Sultan et al., 2016] Sultan, M. A.; Castelli, V. & Florian, R. (2016). A joint model for answer sentence ranking and answer extraction. *TACL*, 4:113--125.

[Tan et al., 2016] Tan, M.; dos Santos, C. N.; Xiang, B. & Zhou, B. (2016). Improved representation learning for question answer matching. Em *54th Annual Meeting of the Association for Computational Linguistics*.

[Tan et al., 2015] Tan, M.; Santos, C. d.; Xiang, B. & Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

[Upbin, 2013] Upbin, B. (2013). Ibm's watson gets its first piece of business in healthcare. `http://www.forbes.com/sites/bruceupbin/2013/02/08/ibms-watson-gets-its-first-piece-of-business-in-healthcare/#1c61087c44b1`. Acessado em: 2016-07-14.

[Voorhees et al., 1999] Voorhees, E. M. et al. (1999). The trec-8 question answering track report. Em *Trec*, volume 99, pp. 77--82.

[Weissenborn et al., 2017] Weissenborn, D.; Wiese, G. & Seiffe, L. (2017). Making neural QA as simple as possible but not simpler. Em *21st CoNLL Conference on Computational Natural Language Learning*, pp. 271--280.

[Weston et al., 2014] Weston, J.; Chopra, S. & Adams, K. (2014). #tagspace: Semantic embeddings from hashtags. Em *2014 EMNLP Conference on Empirical Methods in Natural Language Processing*, pp. 1822--1827.

[Woods & Kaplan, 1977] Woods, W. A. & Kaplan, R. (1977). Lunar rocks in natural english: Explorations in natural language question answering. *Linguistic structures processing*, 5:521--569.

[Yang et al., 2017] Yang, Z.; Hu, J.; Salakhutdinov, R. & Cohen, W. W. (2017). Semi-supervised QA with generative domain-adaptive nets. Em *55th ACL Annual Meeting of the Association for Computational Linguistics*, pp. 1040--1050.

[Yin et al., 2016] Yin, W.; Yu, M.; Xiang, B.; Zhou, B. & Schütze, H. (2016). Simple question answering by attentive convolutional neural network. *arXiv preprint arXiv:1606.03391*.

[Yosinski et al., 2014] Yosinski, J.; Clune, J.; Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks? Em *Annual Conference on Neural Information Processing Systems*, pp. 3320--3328.

[Yu et al., 2016] Yu, Y.; Zhang, W.; Hasan, K. S.; Yu, M.; Xiang, B. & Zhou, B. (2016). End-to-end reading comprehension with dynamic answer chunk ranking. *CoRR*, abs/1610.09996.