Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

O paradigma de supervisão distante para extração de relações

Leonardo Martinez Ikeda

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Prof. Dr. Denis Deratani Mauá

São Paulo
17 de Dezembro de 2021

Resumo

Leonardo Martinez Ikeda. **O paradigma de supervisão distante para extração de relações**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

A extração de relações é uma das tarefas da extração de informações, cujo objetivo consiste em extrair relações semânticas entre entidades a partir de textos não estruturados. Por exemplo, da sentença de texto "Brasília é a capital federal do Brasil", é possível perceber que as entidades *Brasília* e *Brasil* estão semanticamente relacionadas. A extração de relações é a tarefa responsável por extrair relações como essa de forma estruturada. Um dos grandes problemas encontrados ao realizar essa tarefa está no fato de que a produção de uma vasta quantidade de dados manualmente rotulados é um processo lento e custoso. Em uma tentativa de lidar com esse e outros problemas, um novo paradigma de aprendizado, chamado de supervisão distante, foi recentemente introduzido. No paradigma de supervisão distante, as informações de uma base de conhecimento externa são alinhadas heuristicamente com diversas sentenças de textos não estruturados para a obtenção dos dados de treinamento. Como a heurística utilizada pode ser violada, embora essa metodologia consiga produzir um vasto conjunto de dados rotulados, parte desses dados possuem rótulos ruidosos, fazendo com que uma série de cuidados precisem ser tomados para reduzir o impacto negativo desses ruídos sobre o desempenho de modelos na tarefa. Este trabalho visa produzir um texto introdutório à tarefa de extração de relações usando supervisão distante, discutindo as ideias centrais por trás do paradigma de supervisão distante, destacando algumas de suas limitações assim como melhorias alcançadas pelos avanços recentes nessa tarefa.

Palavras-chave: processamento de linguagem natural. extração de relações. supervisão distante.

Abstract

Leonardo Martinez Ikeda. **The distant supervision paradigm for relation extraction**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2021.

Relation extraction is a task of information extraction, whose goal consists of extracting semantic relationships between entities from unstructured text. For example, from the sentence "Brasília is the federal capital of Brazil", it can be seen that the entities *Brasília* and *Brazil* are semantically related. Relation extraction is the task responsible for extracting such relations in a structured manner. One of the big problems faced when tackling this task is that producing vast amounts of manually labeled data is a slow and expensive process. In an attempt to deal with this and other problems, a new learning paradigm, called distant supervision, was recently introduced. In the distant supervision paradigm, information from an external knowledge base are heuristically alligned with various unstructured text sentences in order to generate the training data. Since the heuristic being used can be violated, although this method produces a large amount of labeled data, part of this data has noisy labels, and a number of precautions must be taken in order to minimize the negative impacts of noisy data over models performance in this task. This work aims to produce an introdutory text to the relation extraction task using distant supervision, discussing the core ideas behind the distant supervision paradigm, highlighting some of its limitations as well as a number of improvements achieved by recent developments in this task.

Keywords: natural language processing. relation extraction. distant supervision.

Lista de Figuras

Sumário

Capítulo 1

Introdução

Atualmente, textos não estruturados compõem grande parte de todo o conteúdo produzido, disponibilizado e armazenado a cada instante na *Web*. No interior desse conjunto de textos, existe um grande volume de conhecimento valioso e cujo acesso é comprometido pela falta de estrutura dos textos. Com o interesse em obter essas informações de forma mais bem estruturada, a tarefa de extração de relações tem ganhado uma atenção cada vez maior nos últimos tempos.

A extração de relações é uma subtarefa da extração de informações, a qual é definida como uma tarefa que visa extrair dados estruturados a partir de um conjunto de documentos não estruturados. Desta forma, tais informações tornam-se facilmente processáveis por máquinas, permitindo que possam servir de base para diversos tipos de aplicações, como sistemas de *question answering* ou de recuperação de informações.

Na tarefa de extração de relações, estamos preocupados, especificamente, em extrair relações semânticas entre entidades presentes em textos. Entendemos por entidade um objeto real ou conceitual que pode ser unicamente identificado. Por exemplo, na sentença "*Mark Zuckerberg* é um dos cofundadores do *Facebook*", é possível notar que uma relação de *cofundador* é expressada entre as entidades destacadas. Por se tratar de uma tarefa complexa, os esforços para realizá-la tem se concentrado em utilizar abordagens baseadas em aprendizado de máquina.

Até recentemente, as abordagens existentes para encarar essa tarefa podiam ser divididas em três tipos: supervisionadas, não supervisionadas e semi-supervisionadas. Cada uma dessas abordagens possui diferentes vantagens e limitações. Abordagens supervisionadas obtêm bons resultados, mas dependem de uma base de sentenças rotuladas para o treinamento. Esse fator limita a quantidade de dados disponíveis, já que a produção de dados de treinamento rotulados para a extração de relações é um processo extrememamente lento e custoso. Além disso, essa limitação faz com que os classificadores obtidos costumem apresentar problemas de sobreajuste em relação ao domínio dos textos utilizados para o treinamento. Abordagens não supervisionadas e semi-supervisionadas, por sua vez, embora possam utilizar um grande volume de dados com pouco esforço, produzem resultados piores, com menor precisão ou em formatos menos estruturados do que se espera.

Diante desse cenário, Mintz *et al.* (2009) introduziram um novo paradigma de aprendi-

zado para essa tarefa, denominado supervisão distante. A principal ideia por trás desse paradigma está em explorar uma base de conhecimento externa como fonte de supervisão, fazendo um alinhamento heurístico dessa base com sentenças de texto, de forma a produzir um conjunto de dados rotulados de larga escala. Feito isso, tal conjunto é utilizado para treinar um classificador. O objetivo disso está em obter uma abordagem capaz de combinar os benefícios dos diferentes tipos de abordagens previamente mencionadas.

Desde sua introdução, grande parte dos trabalhos de extração de relações têm se utilizado desse novo paradigma com sucesso, obtendo resultados cada vez mais promissores. Atualmente, o estado da arte na extração de relações sob o paradigma de supervisão distante está em modelos baseados em redes neurais. O primeiro desses modelos foi apresentado no trabalho feito por Zeng, Liu, Y. Chen *et al.* (2015), utilizando uma modelagem baseada em redes neurais convolucionais. Desde então, uma série de melhorias a esse modelo convolucional (Y. Lin *et al.*, 2016, Wu *et al.*, 2019), assim como a utilização de outras abordagens baseadas em redes neurais (P. Zhou *et al.*, 2016, Xu e Barbosa, 2019, Zhu *et al.*, 2020) têm sido os principais caminhos tomados por pesquisadores nessa tarefa.

Neste texto, estudaremos a fundo algumas das principais abordagens da bibliografia, com foco especial para duas delas: o modelo original apresentado por Mintz *et al.* (2009) ao introduzir o novo paradigma de aprendizado, assim como o primeiro modelo baseado em redes neurais convolucionais proposto por Zeng, Liu, Y. Chen *et al.* (2015). O objetivo deste trabalho é a elaboração de um texto introdutório à tarefa de extração de relações sob o paradigma de supervisão distante, e assim sendo, esses modelos constituem bons exemplos dos caminhos que podem ser tomados para abordar essa tarefa. Além da descrição mais detalhada desses dois modelos, também apresentaremos de forma simplificada as ideias por trás de outras abordagens importantes, seguintes a esses modelos, e que buscaram tratar de algumas das deficiências provenientes da adoção desse novo paradigma.

Este texto está estruturado da seguinte forma: no Capítulo 2, apresentaremos todos os conceitos fundamentais necessários para o entendimento do que será discutido neste trabalho. No Capítulo 3, introduziremos o paradigma de supervisão distante para extração de relações, fazendo isso por meio de algumas das primeiras abordagens elaboradas na literatura. No Capítulo 4, damos foco a algumas das primeiras e mais influentes abordagens baseadas na utilização de redes neurais convolucionais. Por fim, no Capítulo 5, concluímos esse trabalho fazendo uma breve revisão da discussão desenvolvida ao longo deste texto, com algumas considerações críticas acerca do trabalho desenvolvido.

Capítulo 2

Conceitos Fundamentais

Neste capítulo, introduziremos alguns conceitos essenciais para o entendimento dos estudos desse trabalho. Primeiro, definimos precisamente a tarefa de extração de relações. Em seguida, explicamos a teoria básica por trás de algoritmos de aprendizado de máquina, com foco especial para conceitos relacionados ao aprendizado supervisionado e redes neurais. Depois, descrevemos tarefas relacionadas de processamento de linguagem natural, que costumam aparecer em modelos de extração de relações na forma de *features*. Por fim, definimos o que é uma base de conhecimento, parte essencial do paradigma de supervisão distante.

2.1 Extração de relações

Antes de falar sobre a tarefa de extração de relações, é preciso ter um entendimento mais claro dos conceitos de entidade e relacionamento, além da definição precisa de relação. Esses conceitos foram introduzidos por P. P. S. Chen (1976), com definições pouco rigorosas. Uma entidade é definida como algo que possa ser unicamente identificado. São exemplos de entidades pessoas, companhias, lugares ou até mesmo eventos. Um relacionamento nada mais é do que uma associação entre duas entidades. Para nossa discussão, o conceito de relação é mais interessante do que o conceito de relacionamento.

Uma relação pode ser entendida como uma forma estruturada de representar a expressão de um relacionamento. Formalmente, definimos uma relação como uma tripla $(r,\ e_1,\ e_2)$, em que e_1 e e_2 são entidades e r um relacionamento entre e_1 e e_2.

A extração de relações é uma tarefa da área de processamento de linguagem natural, cujo objetivo está em, a partir de um *corpus* de sentenças de texto com entidades reconhecidas, extrair relações semânticas entre essas entidades em um formato estruturado. Em processamento de linguagem natural, um *corpus* nada mais é do que uma coleção de documentos textuais.

Para exemplificar melhor a tarefa, considere a seguinte sentença:

> "*Bill Gates* foi um dos cofundadores da *Microsoft*."

Nela, temos a presença das entidades *Bill Gates* e *Microsoft*. Semanticamente, é possível reconhecer também a existência do relacionamento de *cofundador* entre essas entidades. Nesse contexto, um modelo que executa a tarefa de extração de relações deve ser capaz de inferir que a sentença acima expressa a relação (*cofundador, Bill Gates, Microsoft*) e extrair essa informação.

Um ponto importante está na categorização do problema. O primeiro tipo de categorização diz respeito à formulação do problema — ou seja, em como estão definidas a entrada e a saída do problema. Seguindo parcialmente a formulação enumerada por Han *et al.* (2020), destacamos três categorias de extração de relações: extração de relações em nível de sentença, extração de relações em nível de *bag* e extração de relações em nível de documento.

A extração de relações em nível de sentença é a configuração mais clássica e intuitiva do problema. Nela, a entrada é uma sentença com as entidades anotadas e como saída temos qual é a relação expressa pela sentença. Na extração de relações em nível de *bag*, por outro lado, a entrada não é cada sentença individual, mas sim vários *bags*, separados por par de entidades. Entende-se por *bag* uma simples coleção de objetos — no nosso caso, cada *bag* é formado por todas as sentenças contendo o par de entidades que define o *bag*. A saída ainda é a relação expressa entre as entidades, mas com base no *bag* como um todo, e não em cada sentença individual. Como veremos no Capítulo 3, essa segunda formulação faz mais sentido com a aplicação do paradigma de supervisão distante ao problema de extração de relações. Por fim, temos a extração de relações em nível de documento, cujo foco está em extrair relações que só podem ser observadas considerando o contexto por trás de um documento de texto como um todo. A entrada nessa formulação é composta por documentos de texto, formados por várias sentenças com as entidades anotadas, enquanto que a saída é um grafo definido pelas diversas relações expressas no documento todo, em que as entidades são os vértices e os relacionamentos são as arestas.

A extração de relações pode ainda ser categorizada entre fechada ou aberta. Na extração de relações fechada, o conjunto de relacionamentos em que estamos interessados é conhecido *a priori* (ou seja, na definição de relação, adota-se a restrição $r \in R$, onde R é o conjunto de relacionamentos conhecidos). Na extração de relações aberta, não há tal restrição, fato que aumenta consideravelmente a complexidade do problema. Neste texto, falaremos em detalhes somente sobre a extração de relações fechada, que é o foco dos modelos da produção bibliográfica que estudaremos e, também, a categoria que tem recebido maior atenção por publicações nessa área.

2.2 Aprendizado de máquina

Uma primeira abordagem para tentar realizar a extração de relações pode ser por meio da identificação de padrões. Por exemplo, a partir de uma sentença como "*Linus Torvalds* é casado com *Tove Torvalds*", podemos pensar em um padrão que extrai uma relação do tipo *cônjuge* entre entidades e_1 e e_2 sempre que um trecho de texto na forma "e_1 é casado com e_2" for encontrado. Entretanto, não é difícil notar que uma abordagem desse tipo é muito trabalhosa e pouquíssima escalável, uma vez que, para obter um bom desempenho quando aplicada a um grande conjunto de textos e relacionamentos, exigiria uma quantidade

enorme de padrões. Por isso, assim como ocorre para muitas das demais tarefas da área de processamento de linguagem natural, as abordagens desenvolvidas para encarar a extração de relação se baseiam em aprendizado de máquina.

O aprendizado de máquina é um campo de estudo no qual são estudados algoritmos capazes de aprender por meio de dados. Um algoritmo de aprendizado de máquina é capaz de observar um conjunto de dados e, com base neles, construir um modelo, que pode ser utilizado para fazer predições a partir de dados novos. Por exemplo, um algoritmo desse tipo pode ser usado para construir um modelo que classifica flores em suas diferentes espécies, a partir de dados como o tamanho de sépalas e pétalas de flores já coletadas. Quando os mesmos tipos de dados de uma flor cuja espécie é desconhecida forem coletados, podemos utilizar esse modelo para tentar prever qual é a espécie dessa flor.

Tipicamente, o aprendizado é dividido em três tipos: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço. Essa nomenclatura diz respeito ao tipo de *feedback* usado no aprendizado (Russell e Norvig, 2010).

No aprendizado supervisionado, o conjunto de dados fornecidos como entrada são rotulados com a saída correta. O algoritmo aprende uma fórmula capaz de fazer o mapeamento dos valores de entrada para os rótulos de saída fornecidos. O exemplo anterior é um exemplo típico de aprendizado supervisionado: os dados das flores podem conter suas características e um rótulo especificando sua espécie, e busca-se um modelo (classificador) capaz de fazer o mapeamento de um vetor contendo os dados de características de flores para os rótulos que correspondem às diferentes espécies.

No aprendizado não-supervisionado, nenhuma informação em relação à saída é dada. Um exemplo clássico em que esse tipo de abordagem é utilizado está na tarefa de agrupamento. O algoritmo de aprendizado separa os dados em diferentes classes de acordo com padrões identificados. Por exemplo, a partir de um conjunto de imagens de cães e gatos, o algoritmo pode ser capaz de capturar padrões como o formato do focinho ou o tipo de pelagem e separar as imagens em categorias distintas.

No aprendizado por reforço, o aprendizado ocorre por meio de recompensas ou punições. Isso costuma ser feito por meio de alguma nota associada às saídas produzidas. Esse tipo de abordagem é útil para o aprendizado de jogos, devido ao fato de que os resultados costumam ser facilmente adaptados para essa abordagem. Em uma partida de jogo da velha, por exemplo, só temos três possíveis resultados: vitória, derrota ou empate. Nessa abordagem, podemos atribuir uma nota positiva para a vitória, negativa para a derrota e neutra para um empate.

Neste texto, só entraremos em mais detalhes em alguns aspectos do aprendizado supervisionado. Nos modelos que veremos mais a fundo, utiliza-se um outro paradigma de aprendizado, ao qual se deu o nome de supervisão distante. Esse paradigma pode ser visto como uma abordagem de supervisão na qual os rótulos possuem ruídos, havendo incerteza acerca de sua veracidade. Em função disso, é uma abordagem que se aproxima muito do aprendizado supervisionado, e por isso, alguns dos conceitos desse tipo de abordagem são essenciais.

Em particular, faremos uma descrição mais focada na teoria e nos conceitos básicos por trás de algoritmos de aprendizado supervisionado do que em detalhes do funcionamento

de algum deles em específico. Assim, em vez de descrevermos como funciona a regressão linear ou árvores de decisão, detalharemos os fundamentos universais por trás desses diversos algoritmos, dando uma visão geral sobre seu funcionamento. Descreveremos, também, alguns dos métodos de avaliação dos modelos de classificação que costumam ser mais utilizados no desempenho de abordagens que realizam a extração de relações.

2.2.1 Aprendizado supervisionado

Como descrito anteriormente, o aprendizado supervisionado é um tipo de abordagem para o problema de aprendizado. No aprendizado supervisionado, os dados de entrada são acompanhados de rótulos que descrevem uma saída correta para aquele conjunto de dados. Com base nas descrições feitas por Abu-Mostafa *et al.* (2012) e Russell e Norvig (2010), formalizamos o problema de aprendizado supervisionado da seguinte forma:

> Dados um conjunto de dados de treinamento $\mathcal{D}$ com N exemplos na forma
>
> $$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$$
>
> que foram obtidos de uma fórmula desconhecida $y = f(x)$, e um conjunto de hipóteses $\mathcal{H}$, encontrar uma fórmula $g : \mathcal{X} \rightarrow \mathcal{Y}$, com $g \in \mathcal{H}$, que se aproxima da fórmula verdadeira desconhecida $f : \mathcal{X} \rightarrow \mathcal{Y}$ (onde os pares $(\mathbf{x}_i, y_i)$ correspondem, respectivamente, à i-ésima entrada e rótulo de $\mathcal{D}$, $\mathcal{X}$ corresponde ao espaço de entrada e $\mathcal{Y}$ ao espaço de saída).

Ou seja, um algoritmo de aprendizado tem a tarefa de encontrar uma fórmula hipótese g que, com base no conjunto de dados de treinamento, faz um mapeamento dos dados de entrada para os rótulos fornecidos. Utilizando como exemplo o problema de classificação de espécies de flores descrito na introdução dessa seção, temos que o conjunto de dados $\mathcal{D}$ é composto por pares contendo as *features* (características) e rótulos de flores na forma $(\mathbf{x}_i, y_i)$. Ou seja, $\mathbf{x}_i$ contém as características da i-ésima observação, tais como tamanho das sépalas e pétalas, enquanto que y_i contém o respectivo rótulo (a espécie da flor) dessa observação. Os espaços de entrada $\mathcal{X}$ e de saída $\mathcal{Y}$ são, respectivamente, todos os possíveis conjuntos de valores das características das flores e todas as possíveis espécies de flores sendo consideradas. Por fim, f é a fórmula que corretamente mapeia as características para a espécie das flores.

Nesse processo, podemos destacar dois passos fundamentais: a escolha de um bom conjunto de hipóteses $\mathcal{H}$, e a escolha da melhor hipótese g dentro do conjunto escolhido. A escolha de um bom conjunto de hipóteses pode ser vista como a escolha do algoritmo de aprendizado, e um dos aspectos mais importantes dessa escolha está no *trade-off* entre viés e variância. De forma simplificada, isso pode ser visto como o equilíbrio entre a escolha de uma hipótese de baixo viés, capaz de descrever bem os dados de treinamento, e uma hipótese de baixa variância, capaz de generalizar bem para dados além dos de treinamento. Outros termos muito utilizados para descrever isso são subajuste e sobreajuste: dizemos que a hipótese está se subajustando aos dados quando não consegue capturar padrões presentes nos dados, e que está se sobreajustando quando passa a capturar padrões específicos dos dados, deixando de generalizar bem para além disso. A Figura 2.1 ilustra bem como isso pode ocorrer, por meio do ajuste de polinômios de variados graus a amostras com ruído

obtidas de uma função senoidal.

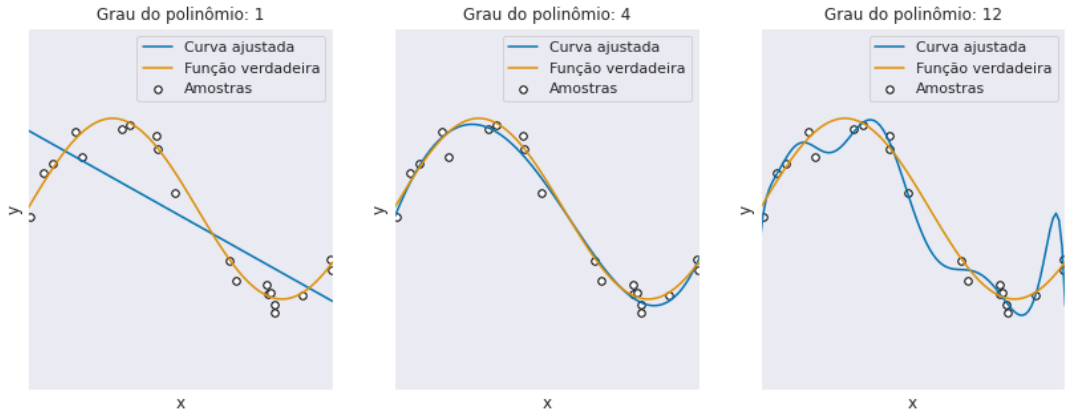

Figura 2.1: *Ajuste de diferentes hipóteses polinomiais aos dados. O polinômio de grau 1 é incapaz de capturar padrões da função verdadeira (subajuste). O polinômio de grau 12, embora se aproxime muito bem à amostra, não se aproxima bem da função verdadeira (sobreajuste).*

Como podemos ver, uma hipótese pode se ajustar muito bem aos dados de treinamento mas, apesar disso, não generalizar bem para além desses dados. A medida da qualidade de uma hipótese, portanto, não é baseada no quão bem ela lida com os dados do conjunto de treinamento, mas sim com dados que não foram observados por ela. Justamente com esse propósito, costuma-se separar todo o conjunto de dados em duas partes - o conjunto de treinamento, utilizado na etapa de treinamento do algoritmo, e o conjunto de teste, utilizado para fazer a avaliação da hipótese final. Na etapa de treinamento, é feita a escolha da melhor hipótese dentro de $\mathcal{H}$. É nesta etapa em que ocorre o aprendizado por parte do algoritmo, em um processo de otimização baseado na minimização de alguma função de perda. Formalmente, uma função de perda $L(y, \hat{y})$ descreve a utilidade perdida ao predizer $\hat{y}$ quando o correto é y. Por exemplo, uma função de perda simples para o nosso problema de classificação de espécies de plantas pode ser a taxa de erro:

$$L(y, \hat{y}) = \begin{cases} 0, & \text{caso } y = \hat{y} \\ 1, & \text{caso } y \neq \hat{y} \end{cases}$$

Apesar de ser uma formulação simples e intuitiva, muitos casos necessitam de uma elaboração melhor. Até agora, nossa discussão se restringiu a problemas de classificação, em que os valores da saída são discretos. Se estivermos lidando com um problema de regressão, em que as variáveis de saída são valores contínuos, esse tipo de formulação é insuficiente. Além disso, podemos querer atribuir pesos diferentes a diferentes tipos de erros. De qualquer forma, do ponto de vista teórico, uma vez definida exatamente qual será a função de perda, o algoritmo de aprendizado deve escolher uma hipótese que minimiza a perda, com base em todos os exemplos observados nos dados de treinamento.

Por fim, é interessante que possamos fazer alguma avaliação da hipótese final, com o intuito de saber se a hipótese final é eficaz para lidar com o problema em questão. Para problemas de classificação, uma das principais ferramentas de avaliação é a matriz de confusão, uma representação na forma de uma tabela dos resultados obtidos por um

classificador. Nela, são apresentadas as contagens de todos os erros e acertos do classificador em relação a cada classe possível. A Tabela 2.1 mostra qual é a aparência de uma matriz de confusão para o problema de classificação binário.

		Valores verdadeiros		
		Positivo	Negativo	
Predição	Positivo	Verdadeiro positivos (VP)	Falso-positivos (FP)	Predições positivas
	Negativo	Falso-negativos (FN)	Verdadeiro negativos (VN)	Predições negativas
		Total de positivos	Total de negativos	

Tabela 2.1: *Matriz de confusão de um problema de classificação binário.*

As células azuis indicam os acertos do modelo, quando a predição bate com o valor verdadeiro, enquanto que as vermelhas indicam os erros, quando a predição difere da realidade. A partir da matriz de confusão, podemos calcular algumas métricas que nos dão diferentes noções do comportamento da hipótese. Aqui, destacaremos três dessas métricas: acurácia, precisão e revocação.

A acurácia é calculada pela fórmula:

$$\text{Acurácia} = \frac{\text{Predições corretas}}{\text{Total geral}} = \frac{VP + VN}{VP + VN + FP + FN}$$

Ou seja, a acurácia nos diz qual é a taxa de acerto geral da hipótese. É uma medida intuitiva e, no geral, boa para avaliar a qualidade de uma hipótese. Entretanto, o problema de classificação pode ser tal que não nos preocupamos apenas com o quanto o modelo acerta ou erra, mas como isso ocorre. Em outras palavras, os tipos de erros e de acertos podem ser mais ou menos importantes de acordo com o problema em questão.

Quanto à precisão, temos a seguinte formulação:

$$\text{Precisão} = \frac{\text{Verdadeiro positivos}}{\text{Predições positivas}} = \frac{VP}{VP + FP}$$

A precisão é uma métrica que nos diz a porcentagem de acerto das predições positivas. Com isso, atribui-se custo maior a falso-positivos, sendo uma boa medida para problemas adequados a casos desse tipo. A classificação de mensagens de *spam* é um desses casos, uma vez que um falso-positivo significa que uma mensagem legítima, potencialmente importante, foi classificada como *spam* e provavelmente não será lida pelo recipiente.

A revocação é calculada da seguinte forma:

$$\text{Revocação} = \frac{\text{Verdadeiro positivos}}{\text{Total de positivos}} = \frac{VP}{VP + FN}$$

Assim, diferente da precisão, a revocação nos diz qual é a taxa de acerto da classe positiva. Essa é uma métrica que atribui maior peso aos falso-negativos. Um caso em que esse tipo de erro pode ser mais importante, por exemplo, está na identificação de alguma doença grave em pacientes, em que um falso-negativo pode significar que um paciente

doente deixará de fazer o tratamento adequado da doença, colocando em risco seu estado de saúde.

Uma forma de juntar esses dois últimos conceitos está na curva de precisão-revocação, que é também a principal forma de avaliar comparativamente os resultados obtidos por modelos de extração de relações na literatura. Em geral, o resultado obtido por algoritmos de classificação é um valor probabilístico, e a inferência final é feita por meio de algum limiar sobre esse valor. Ou seja, feita toda a etapa de treinamento do algoritmo, ao se deparar com uma nova entrada, o algoritmo retorna a probabilidade daquela entrada ser positivamente classificada, e um limiar estabelecido determinará se aquela classificação será considerada ou não. Como é de se imaginar, o algoritmo obterá resultados distintos conforme variamos esse limiar. Limiares altos fazem com que somente classificações de alta confiabilidade sejam consideradas, mas podem acabar fazendo com que muitas classificações potencialmente corretas se percam. Limiares mais baixos, por outro lado, podem ser menos precisos ao mesmo tempo em que cobrem parte maior da classe positiva. É justamente essa contraposição que é exposta pela curva de precisão-revocação.

A curva de precisão-revocação é obtida anotando-se os diferentes valores de precisão e revocação obtidos para diversos limiares entre 0 e 1, e projetando esses valores sobre um gráfico bidimensional, usualmente com os valores da precisão no eixo vertical e os valores de revocação no eixo horizontal. Uma maneira de interpretar essa curva é, como mencionado anteriormente, como uma espécie de *trade-off* entre os falso-positivos e os falso-negativos. A Figura 2.2 apresenta um exemplo desse tipo de representação.

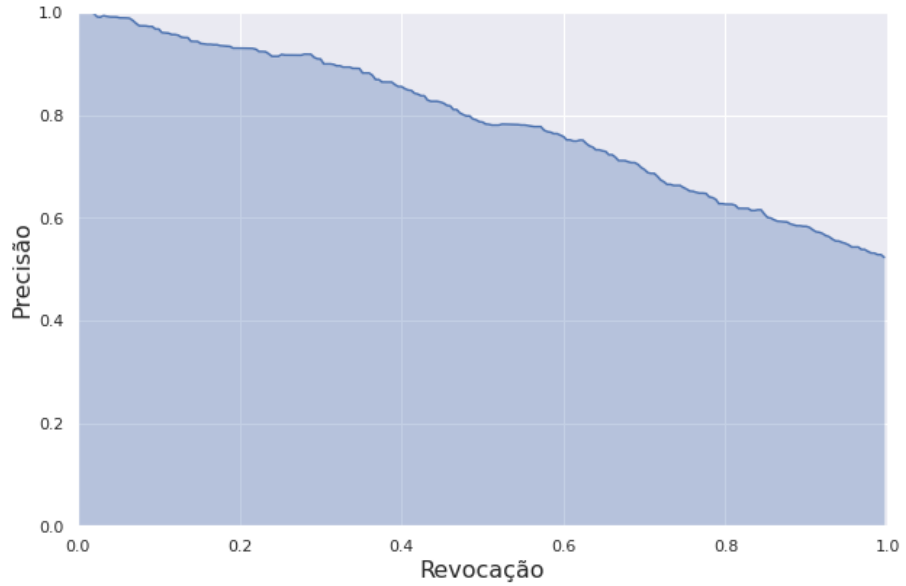

Figura 2.2: *Curva de precisão-revocação.*

2.2.2 Redes neurais

Redes neurais são, atualmente, a família de algoritmos mais representativa do chamado aprendizado profundo, e têm esse nome por serem modeladas com base no circuitos neurais biológicos de cérebros de animais. O nome aprendizado profundo faz referência ao fato

de que esses algoritmos são modelados como várias camadas de circuitos, fazendo com que o caminho tomado ao computar uma saída a partir de dada entrada seja composto por várias etapas, diferente de muitos outros algoritmos de aprendizado.

O componente unitário básico de uma rede neural é um neurônio artificial, inspirado nos neurônios encontrados no cérebro. Um neurônio artificial recebe uma série de valores na entrada, e calcula uma somatória desses valores. Essa somatória é ponderada, e cada um dos valores de entrada recebe um peso diferente, determinado durante a etapa de treinamento. Além disso, um valor de viés, específico para o neurônio e também aprendido durante o treinamento, é adicionado à somatória. Por fim, aplica-se uma chamada função de ativação, que nada mais é do que uma não linearidade responsável por padronizar o valor obtido pela somatória para uma determinada faixa de valores, gerando a saída. Seguindo a nomenclatura baseada no funcionamento de circuitos neurológicos, essa saída costuma receber o nome de ativação. Na Figura 2.3, apresentamos a ilustração de um neurônio artificial com quatro valores de entrada.

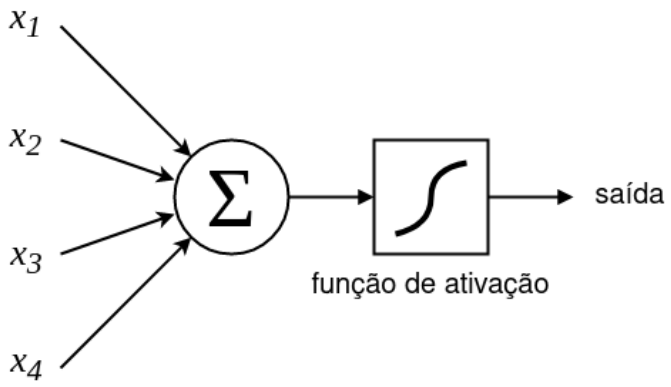

Figura 2.3: *Ilustração de um neurônio artificial com quatro valores de entrada. Os pesos atribuídos a cada valor estão representados de forma implícita em cada arco que liga os valores de entrada ao neurônio. O valor de viés está implicito no próprio neurônio.*

Formalizando o que foi dito, um neurônio artificial recebe como entrada um vetor $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, com n valores, calcula a soma entre o valor de viés b do neurônio e a somatória $\sum_n x_n w_n$, onde w_n corresponde ao peso atribuído à n-ésima entrada, e aplica uma função de ativação f ao resultado obtido com a somatória. O resultado de todos esses cálculos é a saída $f(b + \sum_n x_n w_n)$.

Como dito, redes neurais são modeladas como várias camadas de circuitos. Uma camada de uma rede neural é um conjunto de neurônios artificiais. Tipicamente, uma rede neural é estruturada de forma hierárquica: temos os valores de entrada que são fornecidos a cada neurônio da primeira camada, as saídas de cada neurônio da primeira camada compõem as entradas dos neurônios da segunda camada, e assim sucessivamente, até que se produza a saída da rede neural. Na Figura 2.4, temos como exemplo a estrutura de uma rede neural desse tipo com duas camadas, que produz uma saída binária.

A vantagem mais aparente deste tipo de modelagem está no fato de que ela permite às variáveis interagirem entre si de formas complexas. Uma forma de interpretar o que é feito por redes neurais está em pensar que desejamos extrair padrões complexos a partir

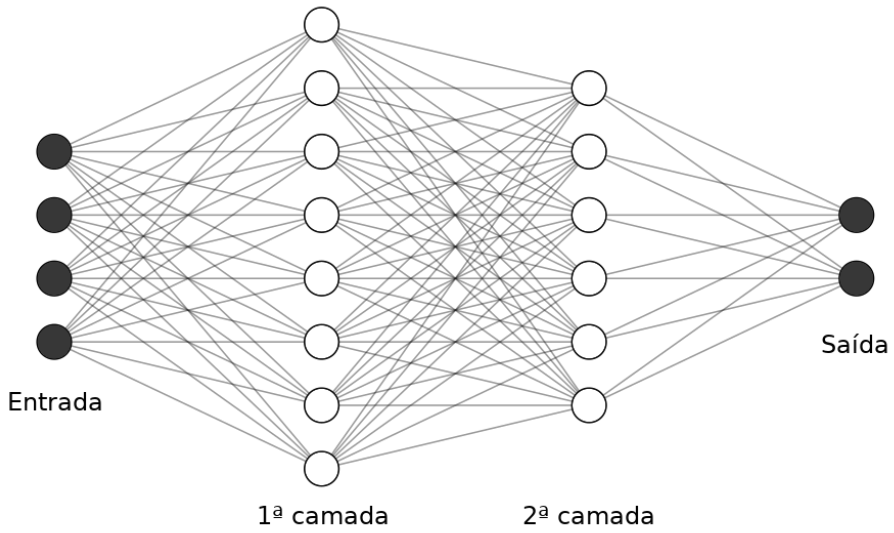

Figura 2.4: *Ilustração da estrutura de uma rede neural com duas camadas, de tamanho 8 e 6, produzindo um resultado binário.*

dos dados, que cada camada é responsável por capturar componentes que expressam determinados padrões, e que tais componentes podem ser divididos em subcomponentes que podem ser capturados pelas camadas seguintes. Os dados de entrada podem ter padrões que determinam as ativações da primeira camada, os padrões das ativações da primeira camada determinam as ativações da próxima camada, e assim por diante. Para que isso ocorra, durante o aprendizado de uma rede neural, os diferentes pesos e vieses associados a cada neurônio são ajustados de forma a otimizar alguma função objetivo de acordo com o resultado obtido. Todo esse processo resulta em um modelo com um poder de expressão muito maior do que muitos dos demais modelos de aprendizado de máquina.

Até o momento, descrevemos uma forma específica de organização das camadas, em que a informação é propagada de trás para frente durante a etapa de ajuste dos pesos, em camadas sucessivas. Redes neurais organizadas dessa forma, em que a informação segue um fluxo unilateral, com neurônios anteriores entregando a informação aos neurônios posteriores, costumam ser chamadas de redes neurais de alimentação direta. Com o desenvolvimento nessa área, diversas outras de organizar as camadas de uma rede neural foram desenvolvidas, e atualmente temos um conjunto de diferentes estruturas de camadas que são comumente utilizadas.

Redes neurais convolucionais

Redes neurais convolucionais possuem dois tipos de camadas fundamentais. A primeira dessas, como o próprio nome sugere, são as camadas de convolução, responsáveis por realizar a operação de convolução sobre os dados. Além desse tipo de camada, temos as camadas de *pooling*, responsáveis principalmente por fazer a redução de dimensionalidade dos dados.

O primeiro passo para entender redes neurais convolucionais está em entender o que é

uma convolução. Formalmente, uma convolução é uma operação envolvendo duas funções, cujo resultado expressa como o formato de uma dessas funções é modificado pela outra. No contexto de redes neurais, costuma ser entendida como a aplicação de um mesmo filtro sobre diferentes pedaços da entrada, produzindo determinado resultado. Quando falamos em processamento de imagens, por exemplo, tanto a entrada (a imagem em questão) como o filtro da convolução podem ser representados por matrizes, e o filtro varre diferentes partes da imagem aplicando uma mesma operação, obtendo a cada passo novos valores que, juntos, formam uma nova matriz como resultado final. Um exemplo desse processo está ilustrado na Figura 2.5. Em redes neurais convolucionais, uma camada de convolução é uma camada em que realizamos diversas operações de convolução, com diversos filtros diferentes. Uma maneira de interpretar isso está em pensar que cada filtro é responsável por tentar achar a ocorrência de algum padrão nos dados. Os filtros utilizados são aprendidos pelo algoritmo durante a etapa de treinamento, e assim é feita a extração de *features* nesses modelos.

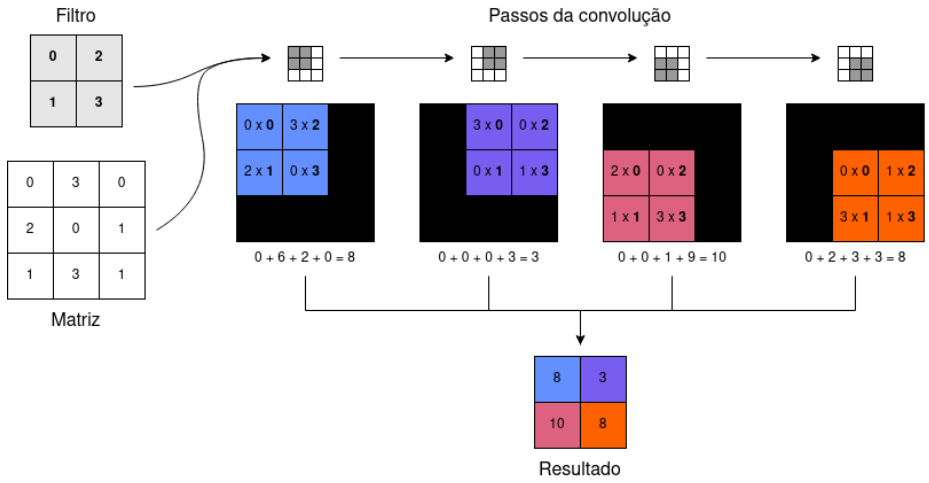

Figura 2.5: *Ilustração da operação de convolução, com uma matriz de tamanho 3x3 e um filtro de tamanho 2x2. O filtro percorre partes da matriz como uma janela móvel, e o resultado é a somatória das diferentes multiplicações dos valores da matriz com os do filtro.*

No processamento de textos, normalmente lidamos com sentenças ou documentos de texto, que podem ser definidos como uma sequência de n palavras $(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$. Também costuma-se utilizar representações vetoriais de palavras, em que cada palavra é representada por um vetor $\mathbf{p} \in \mathbb{R}^d$, de dimensão fixa d. Na Subseção 2.3.4, apresentamos um exemplo de representação vetorial desse tipo. Embora essa representação permita que a entrada possa ser interpretada como uma matriz (por exemplo, considerar que uma sentença é uma matriz em que cada coluna corresponde à representação vetorial de cada palavra), em processamento de textos a convolução costuma aparecer como uma operação linear, em que um mesmo filtro é aplicado a diferentes janelas de palavras. Por exemplo, considere que temos uma sentença $s = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$ definida como uma sequência de n palavras. Com uma janela de tamanho de tamanho k, e m filtros diferentes, temos uma matriz de convolução $C \in \mathbb{R}^{(k \cdot d) \times m}$. Uma operação de convolução consiste no produto entre a concatenação dos vetores das palavras de uma janela com a matriz C. Uma camada de convolução aplica essa operação para todas as janelas possíveis de acordo com o tamanho k escolhido. Por exemplo, utilizando $k = 3$, a primeira dessas janelas é $j_1 = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$, a

segunda janela é $j_2 = [\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$, e assim em diante. Assim, para cada i-ésima janela possível, aplicamos a operação de convolução $F_i = j_i C$, obtendo como resultado um conjunto de i vetores com m *features*. Uma intuição importante nesse processo todo está em pensar que cada filtro é responsável por extrair uma *feature*. No contexto de processamento de textos apresentado, uma *feature* é um valor unitário. Essa intuição facilita a interpretação do resultado obtido por uma convolução: cada filtro extrai um valor que representa uma *feature*, em uma dada janela aplicamos m filtros obtendo um vetor com m *features*, e, como fazemos isso para i janelas, temos como resultado um conjunto de i vetores com m *features*. Esse processo pode ser visto na Figura 2.6.

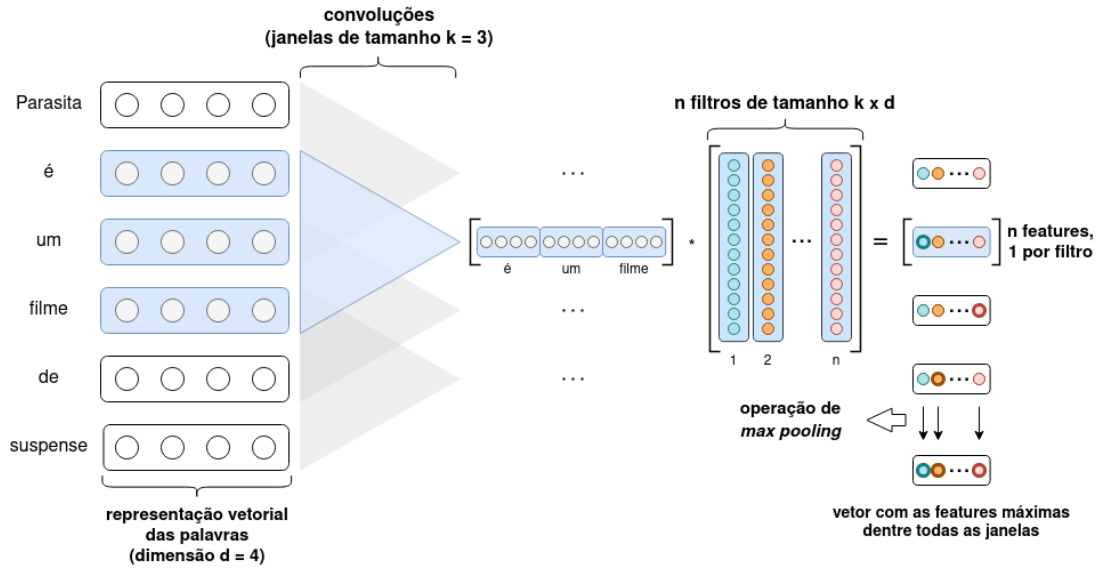

Figura 2.6: *Ilustração de uma abordagem básica de convoluções aplicadas a tarefas de processamento de texto. Cada círculo representa um valor. Os vetores à direita são os resultados das convoluções de cada janela possível.*

Além das camadas de convolução, outro elemento fundamental de redes neurais convolucionais são as camadas de *pooling*. Uma camada de *pooling* é responsável por fazer uma subamostragem. Podem ser entendidas como camadas responsáveis por fazer a sintetização das *features*. Existem diferentes tipos de *pooling* que são comumente utilizados: é muito comum que se use o *max pooling*, no qual são retirados os valores máximos de alguma dimensão. A Figura 2.6 apresenta um bom exemplo desse tipo de operação: no lado direito da figura, temos os vetores de *features* obtidos pela convolução aplicada a cada janela de palavras, em que cada valor desses vetores é obtido por um filtro em específico. A operação de *max pooling* representada na figura se responsabiliza por sintetizar todos esses vetores em um único, selecionando os maiores valores dentre aqueles gerados por um mesmo filtro, representados por círculos com bordas mais grossas.

Entre os principais motivos por trás da utilização de camadas de *pooling*, destacam-se os fatos de que elas produzem saídas de dimensão padronizada e de que são capazes de fazer a redução de dimensionalidade, idealmente mantendo as informações mais importantes. Voltando novamente ao exemplo da figura, isso está bem evidente, uma vez que a operação de *pooling* mostrada é responsável por sintetizar i vetores de *features* em um único vetor, de dimensão fixa m, correspondente ao número de filtros utilizados na convolução. A

redução da dimensionalidade é importante tanto em termos de eficiência computacional, por simplificar eventuais cálculos, como para adequar os resultados para a tarefa em questão — para classificação, por exemplo, é interessante termos uma saída de tamanho fixo, para que ela possa ser fornecida a um classificador. Quanto à capacidade dessa operação de manter as informações mais relevantes, voltando à intuição de que cada filtro é responsável por extrair uma *feature*, podemos interpretar que o valor de uma *feature* nos diz se determinado padrão foi encontrado nos dados. Por exemplo, a sequência de palavras "um filme de" pode ser considerada importante para uma tarefa de extração de relações se quisermos extrair relações que expressam o gênero de filmes de cinema. Nesse caso, é possível que um filtro seja responsável por capturar padrões semelhantes a esse, e que ao passar pela janela de palavras "um filme de", a *feature* extraída por esse filtro tenha um valor alto em comparação ao valor das demais janelas. Dessa forma, ao considerarmos somente os valores máximos obtidos por cada filtro em todas as janelas, espera-se que a perda de informação seja pouco significativa.

A operação de *pooling*, assim como a convolução, é aplicada a janelas. A diferença é que ela costuma ser aplicada a janelas tais que não haja sobreposição de elementos entre uma janela e outra (palavras de uma janela pertencem somente a ela). No exemplo da Figura 2.6, a operação de *pooling* é global, por fazer a operação considerando o texto todo como uma única janela. Assim, a saída obtida com a operação pode ser considerada uma representação vetorial do texto como um todo: isso faz sentido para tarefas em que estamos interessados em aspectos globais do texto, quando a localidade não nos interessa muito (não importa muito onde certo padrão ocorreu, mas sim que ele ocorreu em algum momento). Tanto para a convolução como para o *pooling*, a definição dos parâmetros da janela é um ponto importante na construção de uma rede neural. O tamanho da janela, por exemplo, determina quantas palavras farão parte de uma janela. Também podemos definir o deslocamento de cada janela: qual é o espaçamento entre as palavras a cada passo que define uma nova janela. Outro ponto importante está na utilização ou não de preenchimento, usualmente um vetor de zeros, antes e depois da sentença. O preenchimento é considerado pelas janelas, e muda a dimensão da saída das operações. Usualmente, esses parâmetros costumam ser escolhidos de forma empírica, testando diferentes combinações de valores dos parâmetros e avaliando qual é o melhor resultado obtido considerando as diferentes combinações.

Outras possibilidades de escolha de modelagem incluem opções como usar simultaneamente vários tamanhos de janelas, fazendo um conjunto de operações de convolução e *pooling* para cada tamanho de janela, ou ainda enfileirar diversas operações de convolução e *pooling*, executando-as de forma sequencial. De qualquer forma, como podemos ver, camadas de convolução e de *pooling* são as componentes básicas de uma rede neural convolucional. Enquanto que em redes neurais tradicionais podemos ter camadas de neurônios que vão alimentando as camadas seguintes, em redes neurais convolucionais, um processo semelhante é feito pelas camadas de convolução e de *pooling*. Assim como ocorre para redes neurais tradicionais, costuma-se aplicar alguma função de ativação aos resultados, antes ou após uma camada final de *pooling*.

2.3 Processamento de linguagem natural

O processamento de linguagem natural é uma área da computação preocupada com a manipulação e entendimento das linguagens naturais por meio de computadores. A extração de relações, sendo uma das tarefas dessa área, se utiliza de muitas das técnicas desenvolvidas nesse campo de estudos. Nesta seção, descreveremos algumas das técnicas essenciais que são utilizadas na etapa de pré-processamento dos modelos estudados.

2.3.1 Ligação de entidades

A ligação de entidades, também conhecida pelo nome de resolução de entidades, é a tarefa responsável por fazer a associação correta entre menções de entidades identificadas em textos com as entidades equivalentes de uma base de conhecimento. Na Seção 2.4, descrevemos o que são bases de conhecimento.

Essa tarefa envolve outra tarefa clássica do processamento de linguagem natural: o reconhecimento de entidades nomeadas. O reconhecimento de entidades nomeadas é a tarefa responsável pela identificação de entidades nomeadas em documentos textuais, tais como pessoas, lugares geográficos, instituições, eventos, etc. Na ligação de entidades, além dessa identificação, estamos preocupados em garantir que menções ambíguas sejam devidamente desambiguadas, e corretamente mapeadas para as entidades correspondentes de uma base de conhecimento. A Figura 2.7 traz uma ilustração desse processo.

Em relação à extração de relações, essa tarefa compõe parte essencial do pré-processamento de texto necessário para a aplicação do paradigma de supervisão distante, como veremos no Capítulo 3. Apesar de muitas das abordagens da literatura não precisarem se preocupar com esse fato, por usarem conjuntos de dados prontos e pré-processados, é importante ressaltar que esse é um passo fundamental para uma aplicação real do paradigma.

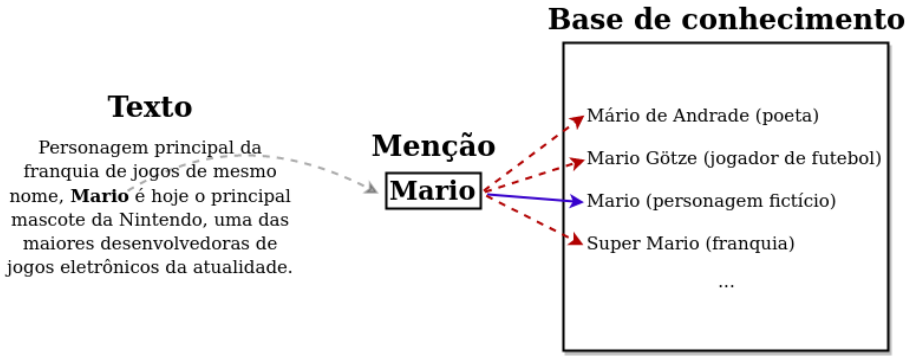

Figura 2.7: *Ilustração da tarefa de ligação de entidades. A menção da entidade, em negrito, deve ser corretamente mapeada para a entidade equivalente da base de conhecimento. A flecha cheia de cor azul indica o mapeamento correto.*

2.3.2 Etiquetagem morfossintática

Como diz o próprio nome, a etiquetagem morfossintática é a tarefa responsável por marcar cada palavra de um texto com sua classe morfossintática correspondente. Na

Figura 2.8, temos um exemplo desse tipo de marcação, aplicado à sentença "O Taj Mahal fica em Agra, cidade indiana". No exemplo, utiliza-se a seguinte notação para as etiquetas: *ADJ* para adjetivos, *ADP* para adposições, *DET* para determinantes, *SPROP* para substantivos próprios, *SUBST* para substantivos comuns, e *VERB* para verbos.

Na linguística, classes de palavras que apresentam comportamento sintático semelhante são agrupadas em classes morfossintáticas. A complicação mais evidente nesse tipo de tarefa está no fato em que uma mesma palavra pode tomar classes morfossintáticas diferentes, causando ambiguidade. Para exemplificar isso, considere a palavra "segundo" nas seguintes sentenças:

"Segundo os pesquisadores, os resultados são promissores"

"O piloto britânico terminou a corrida em segundo lugar"

"O segundo é uma das principais unidades de medida do tempo"

Em cada uma das sentenças, a palavra toma classes morfossintáticas diferentes (preposição, adjetivo e substantivo, respectivamente). Como a identificação de cada classe não segue regras tão claras, mas sim costuma ser feita levando em consideração o contexto por trás das palavras, essa é uma tarefa que não pode ser realizada de maneira trivial por um computador, exigindo a utilização de aprendizado de máquina para isso.

Assim como para muitas outras tarefas da área, a utilização desse tipo de marcação como parte das *features* léxicas é importante para a extração de relações, uma vez que anexa informação adicional às palavras. Como é uma tarefa que está fortemente ligada à língua utilizada, sofre mudanças significativas de acordo com ela. Atualmente, em função do alto desempenho obtido por certas modelagens, alguns consideram que é uma tarefa solucionada, embora essa ideia tenha sido questionada por certos autores, que argumentam que as medidas de desempenho comumente utilizadas podem não ser tão adequadas (Manning, 2011; Giesbrecht e Evert, 2009).

2.3.3 Árvores de dependência

Uma árvore de dependência é um tipo de árvore de análise sintática, que é uma forma de representar a estrutura sintática de uma sentença como uma árvore enraizada, de acordo com alguma gramática. De forma resumida, podemos dizer que uma gramática especifica formalmente a estrutura de determinada linguagem. Árvores de dependência baseiam-se em gramáticas de dependência, nas quais a estrutura sintática é formada por relacionamentos binários entre itens.

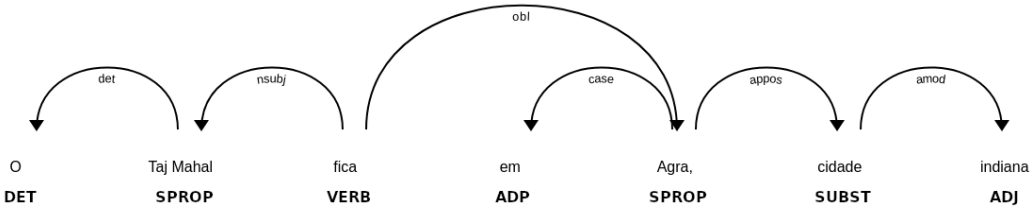

Figura 2.8: *Exemplo da árvore de dependência e etiquetagem morfossintática aplicados a uma sentença. Abaixo das palavras, encontram-se suas classes gramaticais, e os arcos entre as palavras representam os relacionamentos entre elas.*

Formalmente, a árvore de dependência de uma sentença pode ser definida como um grafo orientado acíclico, em que os vértices são as palavras e os arcos os relacionamentos. É uma forma de capturar relacionamentos de dependência entre as palavras. Um exemplo de uma árvore desse tipo está presente na Figura 2.8.

Árvores de análise sintática e árvores de dependência são parte de um extenso campo de estudos de linguística, e por isso, detalhes maiores sobre essas representações, tais como os diferentes tipos de relacionamentos em árvores de análise sintática, fogem do escopo deste trabalho. Para os nossos propósitos, é importante saber que costumam ser usadas na obtenção de *features* sintáticas em modelos de extração de relações, uma vez que fornecem algum entendimento sobre a estrutura sintática de uma sentença.

2.3.4 *Word embeddings*

O termo *word embedding* é comumente utilizado na literatura para fazer referência à representação de palavras por meio dos chamados *embeddings*. Um *embedding* é uma forma de representação de variáveis discretas por meio de vetores de dimensão relativamente baixa e valores contínuos. Assim, um *word embedding* faz o mapeamento de cada palavra de um dado vocabulário (por exemplo, todas as palavras extraídas de um certo *corpus*) para um vetor n-dimensional de valores reais. Essas representações são obtidas por meio da aplicação de abordagens baseadas em aprendizado de máquina, tipicamente em conjuntos de textos grandes, com número de palavras na ordem dos bilhões. Ao final desse processo, temos um vetor para cada palavra distinta encontrada no conjunto de textos. A dimensão desses vetores é a mesma, e depende da modelagem exata utilizada no treinamento.

O objetivo por trás disso está em obter vetores em que algum grau de similaridade é expresso entre palavras. Por exemplo, espera-se que palavras que representam conceitos semelhantes, tais como "gato" e "cachorro", ou "Brasil" e "Argentina", estejam próximas umas das outras no espaço vetorial. O princípio que guia as modelagens usadas para obter esses vetores é o de que o significado de uma palavra pode ser capturado pelo contexto em que ela costuma aparecer, ou seja, pelas palavras que costumam aparecer juntas dela. Um exemplo de como isso pode funcionar bem é dado por D. Lin (1998), com a palavra "tejuino", que é raramente utilizada e por isso costuma ser desconhecida. Apesar disso, por meio de frases como "uma garrafa de tejuino está na mesa", "tejuino te deixa bêbado" e "tejuino é feito a partir do milho", conseguimos inferir que "tejuino" é algum tipo de bebida alcoólica.

Apesar de ser uma ideia simples, na prática, esse tipo de representação de palavras consegue capturar múltiplos graus de similaridade, como mostrado por Mikolov *et al.* (2013). Um exemplo clássico disso dado pelos autores está na capacidade desses vetores em representar relacionamentos entre palavras. Por exemplo, considere que um modelo desse tipo foi treinado, e que os vetores $\mathbf{v}_{\text{homem}}$, $\mathbf{v}_{\text{rei}}$, $\mathbf{v}_{\text{mulher}}$ e $\mathbf{v}_{\text{rainha}}$ são as representações obtidas para as palavras "homem", "rei", "mulher" e "rainha", respectivamente. Podemos utilizar álgebra vetorial para obter um vetor $\mathbf{w} = \mathbf{v}_{\text{rei}} - \mathbf{v}_{\text{homem}} + \mathbf{v}_{\text{mulher}}$, e verificar que, dentre todos os vetores de todas as palavras, o que mais se assemelha ao vetor obtido é $\mathbf{v}_{\text{rainha}}$. Esse resultado indica que relacionamentos são capturados por transformações vetoriais: a transformação de $\mathbf{v}_{\text{homem}}$ para $\mathbf{v}_{\text{rei}}$ é semelhante à de $\mathbf{v}_{\text{mulher}}$ para $\mathbf{v}_{\text{rainha}}$. De fato, na prática isso ocorre para uma série de outros relacionamentos interessantes. As Figuras 2.9 e 2.10

ilustram a propriedade descrita.

Muitas tarefas se beneficiam da utilização desse tipo de representação de palavras. Para a extração de relações, assim como é feito para muitas outras tarefas, costuma-se utilizar *word embeddings* previamente treinados. Esses vetores costumam ser obtidos pelo treinamento em conjuntos de dados extensos, obtidos a partir de fontes como *Wikipedia* ou *Twitter*, e utilizando abordagens conhecidas da bibliografia, como o *word2vec* (Mikolov *et al.*, 2013) ou *GloVe* (Pennington *et al.*, 2014).

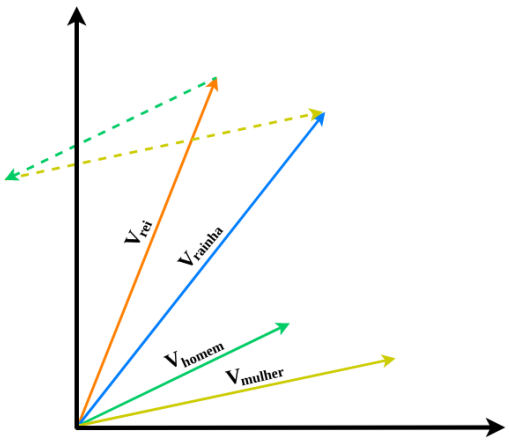

Figura 2.9: *Ilustração de como* ***word embeddings*** *representam relacionamentos entre palavras com álgebra vetorial. As flechas tracejadas indicam translações dos vetores da sua respectiva cor.*

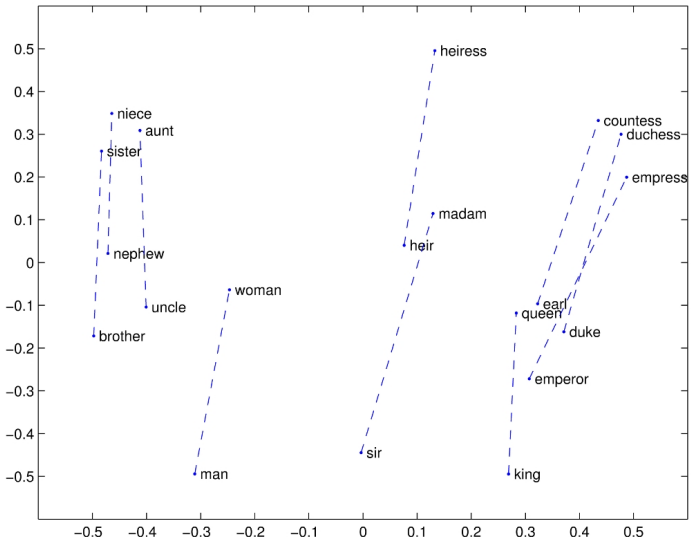

Figura 2.10: *Ilustração de um relacionamento de gênero entre diversas palavras em inglês.* [1].

2.4 Bases de conhecimento

Embora seja difícil encontrar uma definição formal para base de conhecimento na literatura, podem ser entendidas como bancos de armazenamento de conhecimento. Uma base

[1] Fonte: https://nlp.stanford.edu/projects/glove/images/man_woman.jpg

de conhecimento é responsável por armazenar informações e fornecer os meios necessários para a pesquisa, compartilhamento, coleta e utilização dessas informações.

No contexto de extração de relações, costumam ser entendidas como repositórios que armazenam informações de maneira estruturada, geralmente na forma de triplas *(sujeito, predicado, objeto)*, permitindo a criação de extensos grafos direcionados de informação, em que os vértices são os sujeitos e objetos, e os arcos são os predicados (Smirnova e Cudré-Mauroux, 2019). Como é possível notar, triplas nessa forma se assemelham muito às relações como definidas na Seção 2.1. Outra característica importante de bases de conhecimento, especialmente no contexto do paradigma de supervisão distante, é que elas devem permitir que as informações armazenadas sejam facilmente consultadas e extraídas. Dessa forma, o alinhamento heurístico entre base de conhecimento e *corpus*, sobre o qual falaremos em mais detalhes no Capítulo 3, não só é possibilitado como também facilitado.

O *Wikidata* (Vrandečić, 2012) é um exemplo de base de conhecimento desse tipo, no qual os dados são armazenados de maneira estruturada por meio de pares propriedade-valor sobre itens. Por exemplo, o item dessa base que se refere ao planeta Terra possui uma propriedade *ponto mais alto* cujo valor é o item que se refere ao monte *Everest*.

Bases de conhecimento são essenciais para a supervisão distante, uma vez que são usadas como fonte de verdade para a supervisão. Assim como o *Wikidata*, muitas bases de conhecimento em larga escala estão amplamente disponíveis hoje, tais como o *DBpedia* (Lehmann *et al.*, 2015) e *YAGO* (Suchanek *et al.*, 2007). Na área de pesquisa em extração de relações, os principais conjuntos de dados prontos utilizados foram obtidos utilizando bases de conhecimento como essas.

Capítulo 3

Supervisão Distante

Neste capítulo, descreveremos o que é o paradigma de supervisão distante para a extração de relações. Primeiro, definiremos o que é o paradigma e falaremos brevemente sobre o estado da arte da tarefa de extração de relações no momento em que esse paradigma foi introduzido, buscando mostrar os motivos por trás de sua introdução. Em seguida, falaremos sobre quais passos precisam ser tomados para uma aplicação básica desse paradigma, utilizando a arquitetura do primeiro modelo de supervisão distante como exemplo. Por fim, discutiremos algumas das vantagens e limitações da adoção desse tipo de abordagem, apresentando algumas das ideias por trás de modelos seguintes ao inicial que buscaram tratar de tais limitações.

3.1 O paradigma e sua história

No que diz respeito à extração de relações, a introdução do paradigma de supervisão distante se deu por meio do trabalho de Mintz *et al.* (2009). Como apontado pelos autores, até o momento dessa publicação podíamos dividir as principais abordagens de extração de relações em três tipos, de acordo com o paradigma de aprendizado adotado: abordagens supervisionadas, abordagens não supervisionadas e abordagens semi-supervisionadas.

Abordagens supervisionadas baseiam-se na utilização de sentenças de texto manualmente rotuladas com os relacionamentos expressos para guiar o processo de aprendizado (G. D. Zhou *et al.*, 2005; Surdeanu e Ciaramita, 2007). Nesse sentido, o problema de extração de relações é visto como um problema de classificação multiclasse, onde o algoritmo de classificação é responsável por escolher, dentre um conjunto de relacionamentos pré-determinados, qual é o relacionamento expresso por dada sentença de texto. Os principais modelos desse tipo baseavam-se na extração de uma vasta quantidade de *features* léxicas, sintáticas e semânticas das sentenças de texto que compõem o conjunto de dados. A grande desvantagem desse tipo de abordagem está na dependência de dados manualmente anotados, cuja obtenção é um processo extremamente lento e custoso. Em função disso, muitas dessas abordagens utilizavam-se de conjuntos de dados de tamanhos relativamente pequenos, da ordem de dezenas de milhares de exemplos. Além desse problema, como esses conjuntos de dados costumavam ser produzidos por meio da anotação manual de

textos extraídos de um *corpus* em particular, os classificadores resultantes apresentavam a tendência de serem enviesados em relação ao domínio do *corpus* utilizado.

Abordagens não supervisionadas, por outro lado, podem utilizar-se de grandes volumes de dados sem muitos problemas, uma vez que não necessitam de rótulos. As principais abordagens dessa época baseavam-se na extração de sequências de palavras entre as entidades, a partir de um grande conjunto de textos (Shinyama e Sekine, 2006). Feito isso, aplica-se um processo de agrupamento e de simplificação a essas sequências, com o objetivo de juntar em um mesmo grupo as diferentes sequências de palavras que expressam semanticamente um mesmo relacionamento, obtendo assim as relações. O principal problema desse tipo de abordagem está no formato em que os resultados são obtidos, o que dificulta a estruturação da informação extraída. Como dito, as relações obtidas são descritas somente por conjuntos de palavras, e não há nenhum tipo de informação adicional que facilite o mapeamento dessas relações para o formato de alguma base de conhecimento em particular.

Abordagens semi-supervisionadas se fundamentavam no aprendizado por meio de *bootstrapping* (Brin, 1999; Rosenfeld e Feldman, 2007). Trata-se de um método de aprendizado iterativo, em que fornecemos como entrada alguns poucos exemplos que denominamos sementes. Essas sementes podem ser exemplos de sentenças ou padrões que expressam uma relação entre duas entidades em particular, ou até mesmo a própria tripla que define a relação (ou seja, somente a informação de que um relacionamento existe entre duas entidades em particular). A partir dessas sementes, inicia-se o processo iterativo: elas são usadas para identificar ocorrências da relação no *corpus*, tais ocorrências são usadas para extrair novos padrões que expressam a relação, esses padrões podem ser usados para encontrar relações entre novas entidades do *corpus*, e o processo se repete. Assim como para as abordagens não supervisionadas, essas abordagens têm a vantagem de serem facilmente aplicadas a grandes volumes de dados. Apesar disso, os resultados dessas abordagens costumavam ter baixa precisão e sofrer de um problema chamado de desvio semântico, uma espécie de acumulação de erros, em que certos padrões que não expressam de fato a relação passam a ser considerados ao longo das iterações, e o significado semântico da relação vai lentamente sendo alterado pela incorporação desses padrões. Diante do cenário exposto, Mintz *et al.* (2009) introduziram o que chamaram de paradigma de supervisão distante, com o objetivo de conseguir uma abordagem que conseguisse juntar os benefícios desses diferentes tipos de abordagens: ter resultados semelhantes aos de abordagens supervisionadas, tanto em termos de precisão como em relação ao formato da saída, e ao mesmo tempo conseguir utilizar um grande de volume de dados sem depender do demorado processo de rotulação manual.

A supervisão distante pode ser vista como um tipo especial de aprendizado supervisionado. A diferença básica da supervisão distante está no fato de que os dados rotulados utilizados não são produzidos manualmente, mas sim de forma automática, aproveitando-se da informação de uma base de conhecimento externa, que serve como fonte de supervisão dos dados. Para alinhar as informações do *corpus* utilizado com a base de conhecimento externa, utiliza-se uma heurística. No trabalho de Mintz *et al.* (2009), a heurística utilizada baseia-se na seguinte hipótese: se duas entidades participam de um relacionamento na base de conhecimento, então todas as sentenças que contém ambas as entidades expressam esse relacionamento. Por depender de uma base de conhecimento externa e incompleta (ou seja, que não contém todos os fatos do mundo que poderia) para a rotulação, sob o paradigma

de supervisão distante a tarefa de extração de relações pode ser interpretada como a tarefa de expandir uma base de conhecimento existente com mais fatos. Geramos um conjunto de dados utilizando a informação presente na base de conhecimento como fonte de supervisão, e nosso objetivo é obter um modelo que consiga identificar padrões a partir dos dados gerados que possam ser generalizados para além deles, permitindo que relações expressas semanticamente em sentenças de texto, mas que não estejam ainda armazenadas na base de conhecimento possam ser facilmente capturadas por este modelo.

De maneira formal, podemos definir o que é feito na supervisão distante da seguinte forma: dados um conjunto de dados $\mathcal{C}$ e uma base de conhecimento $\mathcal{B}$, na supervisão distante alinhamos cada instância de $\mathcal{C}$ com informações de $\mathcal{B}$ por meio de uma heurística $\mathcal{H}$. Especificamente, para a extração de relações, na supervisão distante temos:

- $\mathcal{E}$ é o conjunto de todas as n entidades $e_i, 1 \leq i \leq n$
- $\mathcal{C}$ é o *corpus* das k sentenças de textos anotadas com pares de entidades $e_a \in \mathcal{E}; e_b \in \mathcal{E}$, na forma $s^k_{(e_a,e_b)}$
- $\mathcal{R}$ é o conjunto de todos os m tipos de relações $r_j, 1 \leq i \leq m$
- $\mathcal{B}$ é uma base de conhecimento de relações (r_j, e_a, e_b), com $r_j \in \mathcal{R}; e_a \in \mathcal{E}; e_b \in \mathcal{E}$
- $\mathcal{H}$ é a heurística proposta por Mintz *et al.* (2009)

Assim, com base em $\mathcal{H}$, para cada par de entidades (e_a, e_b), com $e_a \in \mathcal{E}; e_b \in \mathcal{E}$ tal que exista uma relação $(r_j, e_a, e_b) \in \mathcal{B}$, alinhamos cada uma das t sentenças na forma $s^t_{(e_a,e_b)}$ com o tipo de relação r_j. Portanto, dados como entrada os conjuntos $\mathcal{E}, \mathcal{C}, \mathcal{R}, \mathcal{B}$, a heurística $\mathcal{H}$ de Mintz *et al.* (2009) alinha as informações de $\mathcal{C}$ e $\mathcal{B}$ para obter um conjunto de dados $\mathcal{D}$ em que cada dado é um par de sentenças e rótulos $(s^t_{(e_a,e_b)}, r_j)$.

Apesar de ser uma ideia relativamente simples, ela pode ser considerada um marco na área, uma vez que possibilitou, subitamente e com poucos esforços, a utilização de um volume de dados muito maior do que era até então possível. Para efeito de comparação, enquanto que trabalhos anteriores se utilizavam de conjuntos de dados com apenas cerca de 17000 instâncias de treinamento e 24 tipos de relações, com essa abordagem os autores conseguiram um conjunto de treinamento contendo cerca de 1,8 milhões de exemplos, com 102 tipos de relações e conectando 940000 entidades distintas. Em função disso, como tinham mais exemplos dos quais podiam extrair *features*, o modelo resultante pôde contar com um conjunto grande de *features* mais específicas, obtendo bons resultados.

3.2 Aplicação do paradigma

Nesta seção, descreveremos o passo a passo detalhado para a aplicação do paradigma de supervisão distante na tarefa de extração de relações. Faremos a descrição de uma abordagem mais básica, buscando dar a visão mais geral o possível sobre como isso pode ser feito, utilizando o modelo elaborado por Mintz *et al.* (2009) como exemplo de alguns dos caminhos que podem ser tomados nas decisões mais importantes. Daqui pra frente, chamaremos a abordagem desenvolvida pelos autores de modelo de Mintz.

Como discutido na seção anterior, não é preciso muito para realizar uma aplicação de

fato do paradigma de supervisão distante. Os elementos fundamentais necessários são um grande *corpus* de texto e alguma base de conhecimento externa. Na Seção 2.4, além de descrevermos o que são bases de conhecimentos, apresentamos alguns exemplos de bases de conhecimento disponíveis atualmente. Quanto ao *corpus* escolhido, cabe apontar que, como estaremos interessados em sentenças de texto, é importante que a separação do texto em sentenças seja feita. Para isso, podemos utilizar um *corpus* de sentenças, ou aplicar a um *corpus* de textos métodos responsáveis por fazer a segmentação do texto em sentenças. No modelo de Mintz, os autores utilizaram como *corpus* o *Freebase Wikipedia Extraction*, um *dump* preprocessado de artigos do *Wikipedia* em inglês, em que os textos dos artigos já estão separados em sentenças, e como base de conhecimento a *Freebase*, uma base de conhecimento colaborativa, hoje extinta em função da inclusão e integração de seu conteúdo à *Wikidata*, outra base de conhecimento colaborativa disponível de forma aberta.

Com o *corpus* de sentenças e a base de conhecimento em mãos, o primeiro passo é fazer a ligação de entidades, processo melhor descrito na Subseção 2.3.1. A primeira etapa desse processo é fazer o reconhecimento de entidades nomeadas no *corpus*, e em seguida a ligação de entidades, configurando tudo para o eventual alinhamento das informações da base de conhecimento com o *corpus*. Pouco foco costuma ser dado a ambas essas partes, sendo comum a utilização de ferramentas bem estabelecidas que realizam cada uma dessas etapas. Um ponto potencialmente importante desse processo está em, além de fazer essa ligação entre *corpus* e base de conhecimento, conseguir informações adicionais que podem ser essenciais para a modelagem. No modelo de Mintz, por exemplo, os autores utilizaram o *Stanford Named Entity Tagger* para categorizar as entidades reconhecidas em cinco categorias (pessoa, organização, localização, outro e nenhum), e essas informações eram usadas no modelo como parte das *features*. Outra possibilidade está em pegar informações sobre as entidades da base de conhecimento, se a base de conhecimento utilizada carregar consigo informações desse tipo.

Feito esse processo, podemos prosseguir para a rotulação das sentenças, aplicando de fato do paradigma de supervisão distante. Como dito anteriormente, a rotulação é feita por meio do alinhamento entre informações do *corpus* com informações da base de conhecimento, e tal alinhamento é feito com base em uma heurística. Essa heurística é a mesma descrita na seção anterior, e presume que se duas entidades presentes em uma sentença participam de um relacionamento na base de conhecimento, então essa sentença expressa a relação equivalente entre essas entidades. Na Figura 3.1, temos uma ilustração do processo completo de obtenção de dados de treinamento sob o paradigma de supervisão distante, e que ilustra bem como funciona o processo de rotulação. Como podemos ver, a base de conhecimento armazena a relação (*cofundador*, *Ben Silbermann*, *Pinterest*). Assim, com base na heurística, rotulamos cada uma das sentenças do *corpus* que contém ambas as entidades *Ben Silbermann* e *Pinterest* com o rótulo correspondente ao relacionamento do tipo *cofundador*. No exemplo da Figura 3.1, já podemos também observar a grande limitação por trás da utilização desse tipo de abordagem — os rótulos obtidos são ruidosos. Para um par de entidades, podemos ter sentenças que não expressam o relacionamento encontrado entre elas na base de conhecimento. Esse é o caso da última sentença do *bag* ilustrado na figura, que não expressa semanticamente a relação de *cofundador* entre as entidades. Uma discussão mais detalhada acerca desse tópico será feita na Seção 3.3.

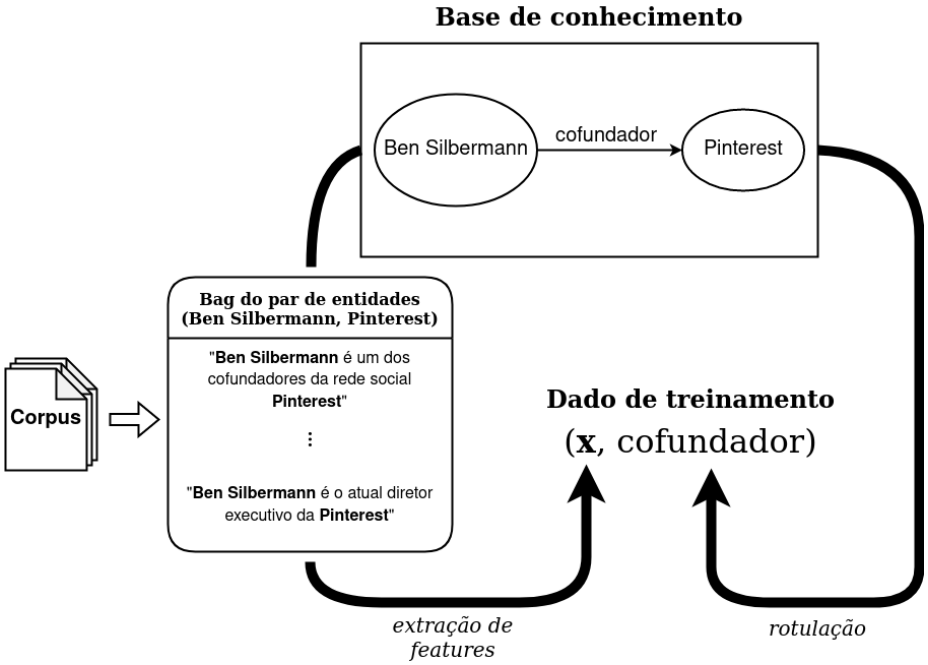

Figura 3.1: *Ilustração do processo de obtenção de dados de treinamento sob o paradigma de supervisão distante.*

Outro ponto importante desse método de rotulação está na obtenção de exemplos negativos. O tipo de alinhamento descrito só funciona para a obtenção de rótulos positivos, e a presença de dados de treinamento negativos é essencial para o desempenho de um classificador. Para a obtenção de dados negativos, costuma-se selecionar amostras de sentenças contendo pares de entidades que não estão relacionados na base de conhecimento e anotá-las com rótulos negativos. Embora seja possível que essas entidades estejam de fato relacionadas, mas que a base de conhecimento não contenha essa informação por estar incompleta, é esperado que com uma base de conhecimento suficientemente grande isso seja pouco frequente, de modo que os ruídos gerados por esses falso negativos não causem um grande impacto no aprendizado.

Continuando o processo, podemos prosseguir para a extração de *features* das sentenças de texto. Um ponto importante relacionado a isso está em qual será a formulação exata do problema que será utilizada. Como descrito na Seção 2.1, temos diversas formas de categorizar a extração de relações segundo a sua formulação. Nas abordagens baseadas em supervisão distante, é muito mais comum que se utilize a formulação de extração de relações em nível de *bag*. Esse tipo de formulação foi introduzido no modelo de Mintz, e o objetivo disso está em se aproveitar do grande volume de dados para obter *features* mais específicas. Ao invés de extrair *features* de cada sentença e considerá-las de forma individual, são extraídas *features* de todas as sentenças pertencentes a um mesmo *bag* e todas elas são combinadas para formar um vetor de *features* muito mais rico. Para a extração de relações, costuma-se fazer o uso de uma combinação de diferentes tipos de *features* léxicas e sintáticas. No modelo de Mintz, as *features* léxicas são compostas por:

- A sequência de palavras entre as entidades

- Um indicador de qual entidade aparece primeiro na sentença
- Uma janela de k palavras à esquerda da primeira entidade, com $k \in \{0, 1, 2\}$
- Uma janela de k palavras à direita da segunda entidade, com $k \in \{0, 1, 2\}$
- Junto de cada uma das palavras acima, suas classes morfossintáticas

Para as *features* sintáticas, os autores utilizaram a árvore de dependência das sentenças para obter:

- Um caminho de dependência entre as duas entidades
- Para cada entidade, um vértice denominado janela, que é adjacente à entidade mas não faz parte do caminho de dependência

O caminho de dependência é o caminho entre as entidades na árvore de dependência. Tomando como exemplo a árvore de dependência da sentença "O Taj Mahal fica em Agra, cidade indiana" mostrada na Figura 2.8, o caminho de dependência entre as entidades *Taj Mahal* e *Agra* é composto pelos arcos (*fica, Taj Mahal*), (*fica, Agra*). Também na figura, podem ser considerados exemplos de vértices janela o vértice *O* para a entidade *Taj Mahal* e os vértices *em* e *cidade* para a entidade *Agra*.

Além dessas *features*, como mencionado anteriormente, os autores também se utilizaram de um categorizador das entidades nomeadas, substituindo as entidades por suas categorias nas *features* finais. Um ponto importante da sua modelagem é que as *features* de cada categoria não são formadas por cada atributo individual listado anteriormente, mas sim pela conjunção deles. Na Tabela 3.1, temos um exemplo das *features* léxicas para a sentença "O Taj Mahal fica em Agra, cidade indiana" seguindo a modelagem dos autores. Cada linha da tabela representa uma *feature* individual, ou seja, cada *feature* léxica é composta por um conjunto de atributos léxicos da sentença. Com os atributos utilizados, a única diferença entre cada *feature* léxica está no valor de k considerado, que define o tamanho das janelas externas às entidades. O mesmo ocorre para as *features* sintáticas citadas anteriormente, e a diferença entre cada uma está nas diferentes combinações de vértices janela utilizados. A adoção das *features* dessa forma conjunta resulta em *features* de alta precisão mas baixa revocação, o que faz sentido para a supervisão distante, por ser uma abordagem que possibilita a utilização de um grande volume de dados. Definido esse processo de extração de *features*, conseguimos extrair um vetor de *features* **x** para cada sentença de texto do nosso conjunto de dados.

Janela esquerda	Classe da entidade 1	Palavras entre entidades	Classe da entidade 2	Janela direita
	Localização	('fica', VERB), ('em', ADP)	Localização	
('O', DET)	Localização	('fica', VERB), ('em', ADP)	Localização	('cidade', SUBST)
('O', DET)	Localização	('fica', VERB), ('em', ADP)	Localização	('cidade', SUBST), ('indiana', ADJ)

Tabela 3.1: *Exemplo de features léxicas da sentença "O Taj Mahal fica em Agra, cidade indiana".*

Para finalizar o processo de obtenção de dados com o objetivo de treinar um algoritmo de classificação, basta dividir os dados em conjunto de treinamento e de teste. Até o momento, tudo o que foi feito gerará um conjunto de dados pronto para ser usado como conjunto de treinamento. Na metodologia usada por Mintz *et al.* (2009), os autores separaram parte das instâncias de cada um dos diferentes tipos de relações obtidas dos dados gerados para

compôr o conjunto de teste. Esse tipo de metodologia também foi utilizado por alguns trabalhos posteriores para obter conjuntos de dados novos, um dos quais veremos na Seção 3.3.

Feito isso, o conjunto de dados está pronto e podemos aplicar um algoritmo de aprendizado supervisionado para realizar a classificação, obtendo os resultados da nossa extração de relações. No modelo de Mintz, os autores se utilizaram de um classificador logístico multiclasse, obtendo para cada par de entidades e respectivo vetor de *features* extraído, um relacionamento entre as entidades e um valor que pode ser interpretado como a confiança do classificador sobre o resultado obtido.

3.3 Vantagens, limitações e trabalhos posteriores

Como dito na Seção 3.1, o paradigma de supervisão distante surgiu com o objetivo de conseguir unir os benefícios das diversas abordagens de extração de relações existente até então. Nesse sentido, podemos dizer que essa abordagem conseguiu cumprir bem seus objetivos. A modelagem baseada em supervisão distante utilizada por Mintz *et al.* (2009) permitiu a utilização de um conjunto de dados muito maior do que era utilizado até então, e como resultado os autores obtiveram um classificador que, segundo sua avaliação, conseguia extrair padrões de alta precisão para um bom número de relações. Na Figura 3.2, temos as curvas de precisão-revocação obtida pelos autores durante a avaliação do modelo, considerando diferentes conjuntos de *features*. A comparação entre modelos anteriores e esse não é tão fácil de se fazer, por usarem conjuntos de dados muito distintos, tanto em relação ao tamanho quanto em relação ao domínio dos textos. De qualquer forma, no trabalho de Surdeanu e Ciaramita (2007), que serve como exemplo de uma das mais recentes abordagens supervisionadas da época do trabalho feito por Mintz *et al.* (2009), os autores relatam terem obtido 74% de precisão e 46.9% de revocação. Embora pela Figura 3.2 seja possível constatar que a abordagem de Mintz *et al.* (2009) só tenha conseguido níveis de precisão comparáveis a esse em níveis de revocação entre de 7,5% e 15%, é possível argumentar que o resultado obtido é mais relevante se considerarmos que o conjunto de dados é cerca de 100 vezes maior. Embora não tenham feito nenhuma constatação empírica disso, Mintz *et al.* (2009) afirmam que, diferente de abordagens supervisionadas de até então, uma das vantagens de sua modelagem está no fato de que ela não sofre de problemas de sobreajuste e dependência de domínio do *corpus*. Além disso, diferente das abordagens não supervisionadas, como utilizamos uma base de conhecimento preexistente para a supervisão, os resultados obtidos já estão no devido formato da base de conhecimento em questão, podendo ser facilmente utilizados.

Apesar de todos esses benefícios, essa abordagem ainda tem problemas consideráveis. A limitação mais óbvia está no fato de que essa abordagem depende da utilização de uma base de conhecimento preexistente e suficientemente grande para seu funcionamento. Nesse sentido, a tarefa de extração de relações pode passar a ser vista como a tarefa de expandir uma base de conhecimento existente, e por isso, é uma abordagem inadequada para construir do zero uma nova base de conhecimento. Como mencionado na seção anterior, ao falarmos sobre a geração de dados com rótulos negativos, dependemos de uma base de conhecimento suficientemente grande para que possamos esperar que não ocorram muitos falso negativos entre os rótulos negativos gerados. Esses falso negativos

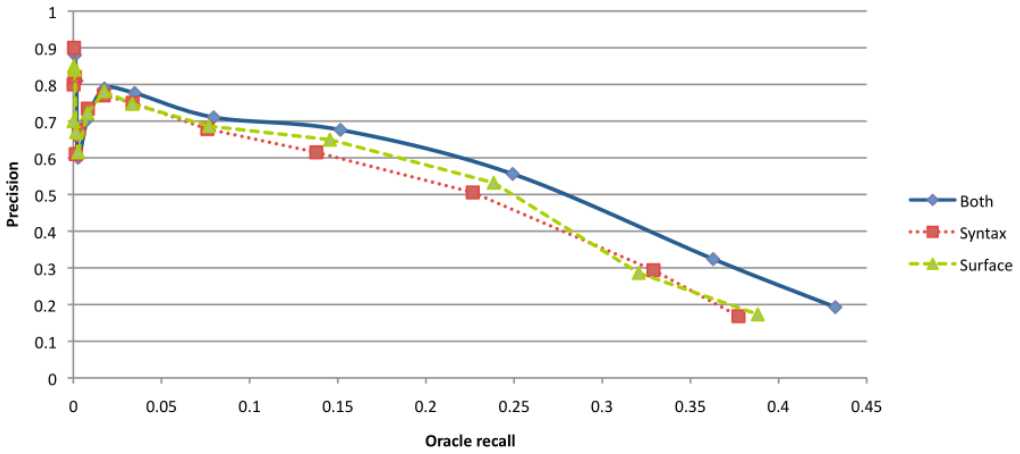

Figura 3.2: *Curva de precisão-revocação, retirada do trabalho de Mintz et al. (2009), mostrando a avaliação do modelo utilizado diferentes conjuntos de features. A linha em verde (Surface) mostra a curva obtida pelo modelo utilizando somente features léxicas. A linha em vermelho (Syntax) se refere ao modelo com somente features sintáticas. Por fim, a linha em azul (Both) se refere ao modelo utilizando ambos os tipos de features.*

também dificultam a avaliação do modelo, uma vez que classificações corretas por parte do classificador podem ser consideradas erradas pela ausência dessa informação na base de conhecimento utilizada. Para compensar isso, alguns trabalhos, incluindo o de Mintz *et al.* (2009), costumam fazer junto da avaliação usual com o conjunto de testes, uma avaliação manual dos resultados obtidos.

Mas além de todos esses problemas, a maior limitação do modelo está no fato de que os rótulos obtidos possuem ruído. Como apontado brevemente ao falar sobre a rotulação na seção anterior, o exemplo dado na Figura 3.1 exemplifica bem isso: a última sentença do *bag* apresentado, "*Ben Silbermann* é o atual diretor executivo da *Pinterest*", não expressa de fato a relação (*cofundador*, *Ben Silbermann*, *Pinterest*), presente na base de conhecimento. A presença de ruídos desse tipo prejudica o aprendizado e, como consequência, o desempenho do algoritmo da tarefa. Voltando ao exemplo da Figura 3.1, por mais que *Ben Silbermann* de fato seja fundador e diretor executivo da mesma empresa, existem diversos fundadores que não são diretores executivos e vice-versa. Assim, a presença do ruído pode fazer com que durante o aprendizado, o algoritmo atribua algum peso positivo ao padrão "é o atual diretor executivo da" para prever a relação de *cofundador* e, ao se deparar com esse mesmo padrão em uma sentença em que o diretor executivo não é também cofundador, possa acabar fazendo essa classificação falso positiva. Como essas rotulações ruidosas ocorrem em função tanto do *corpus* como da base de conhecimento utilizada, a intensidade dos problemas causados pelos rótulos ruidosos muda de acordo com a aplicação.

Essa limitação da abordagem foi percebida rapidamente por pesquisadores da área, e logo já foram elaborados esforços para tratar dela. Dos trabalhos desse tipo, o primeiro a se destacar foi a abordagem de Riedel *et al.* (2010). Assim como observamos, os autores notaram o problema dos rótulos ruidosos, em especial para o caso em que a base de conhecimento e o *corpus* utilizado possuem muito pouca relação. No modelo de Mintz, os autores utilizaram a *Freebase* como base de conhecimento e textos da *Wikipedia* como *corpus*.

Um ponto crucial da *Freebase* é que, além de ser uma base de conhecimento colaborativa, parte dos seus dados eram coletados de fontes externas, sendo o *Wikipedia* uma das principais. Essa proximidade entre base de conhecimento e *corpus* faz com que a heurística original não seja violada tão frequentemente quanto poderia. Os autores constataram isso ao também alinhar a *Freebase* a um *corpus* de textos extraídos do jornal estadunidense *The New York Times*, obtendo um novo conjunto de dados junto do conjunto original obtido por Mintz *et al.* (2009). Feito isso, retiraram de novo e do antigo conjunto de dados 100 amostras das 3 relações mais frequentes, comparando qual a frequência de violação da heurística para cada conjunto. Nessa verificação, viram que a heurística foi violada em média 31% das vezes no novo conjunto de dados, contra apenas 16,66% no anterior. Esse novo conjunto de dados obtido pelos autores permanece sendo até hoje um dos mais utilizados por novos trabalhos de extração de relações via supervisão distante.

A principal contribuição dos autores foi a elaboração de uma nova heurística, que é uma versão relaxada da heurística anterior, definida da seguinte forma: se duas entidades participam de um relacionamento na base de conhecimento, então ao menos uma sentença que contém essas mesmas entidades expressa esse relacionamento. Como é possível notar, é uma abordagem que adiciona uma complicação ao problema, uma vez que agora desconhecemos quais sentenças dentro de um *bag* expressam ou não determinada relação. Para se adequar a essa nova heurística, os autores desenvolveram um modelo de grafo não direcionado, baseando-se no paradigma de aprendizado múltipla instância. No aprendizado múltipla instância, as instâncias de treinamento são divididas em *bags*, e *bags* com ao menos uma instância positiva são rotulados positivamente, enquanto que *bags* sem instâncias positivas são rotulados negativamente. Partindo dos *bags* rotulados, durante o aprendizado buscamos inferir qual é o rótulo de *bags* não vistos, ou qual é o conceito que liga os *bags* positivamente rotulados, com o objetivo de conseguir rotular as instâncias de cada *bag*.

A modelagem gráfica utilizada pelos autores está representada na Figura 3.3. A ideia por trás da modelagem está em modelar tanto as relações entre pares de entidades como as sentenças individuais em que essas entidades aparecem. Para cada par de entidades e_1 e e_2 que aparecem conjuntamente em ao menos uma sentença, uma variável de relação Y denota qual é o tipo da relação entre as entidades. Para cada uma das n sentenças nas quais e_1 e e_2 aparecem conjuntamente, uma variável binária Z_i, com $i \in \{1, 2, \ldots, n\}$, denota se a relação é expressa ou não pela i-ésima sentença. Assim, dado um par de entidades, temos uma modelagem que prediz tanto o tipo do relacionamento entre as entidades como se cada uma das sentenças expressa o relacionamento em questão. Para adequar essa modelagem ao aprendizado de máquina, utilizaram um grafo de fatores, uma abordagem baseada em um grafo bipartido que representa a fatorização de uma função — no caso, a distribuição de probabilidade condicional $p(Y = y, \mathbf{Z} = \mathbf{z}|\mathbf{X})$ (onde y é a atribuição de uma relação em específico, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ é uma matriz de *features*, com cada $\mathbf{x}_i$ contendo *features* extraídas da i-ésima sentença, $\mathbf{Z}$ denota o estado de todas as variáveis Z_i juntas, e $\mathbf{z} = (Z_1, Z_2, \ldots, Z_n)$ é uma atribuição em particular de $\mathbf{Z}$) é definida em função de três tipos de fatores, representados por quadrados na Figura 3.3. O primeiro desses, adjacente somente à variável Y, reflete o viés do modelo a um tipo específico de relação. O segundo, conecta a variável da relação com cada sentença. Por fim, temos os fatores que conectam cada variável Z_i ao vetor de features correspondente $\mathbf{x}_i$. Além disso, os autores se utilizam

de outros mecanismos de aprendizado, sobre os quais não entraremos em detalhes. Para o leitor interessado, recomenda-se a leitura completa do artigo.

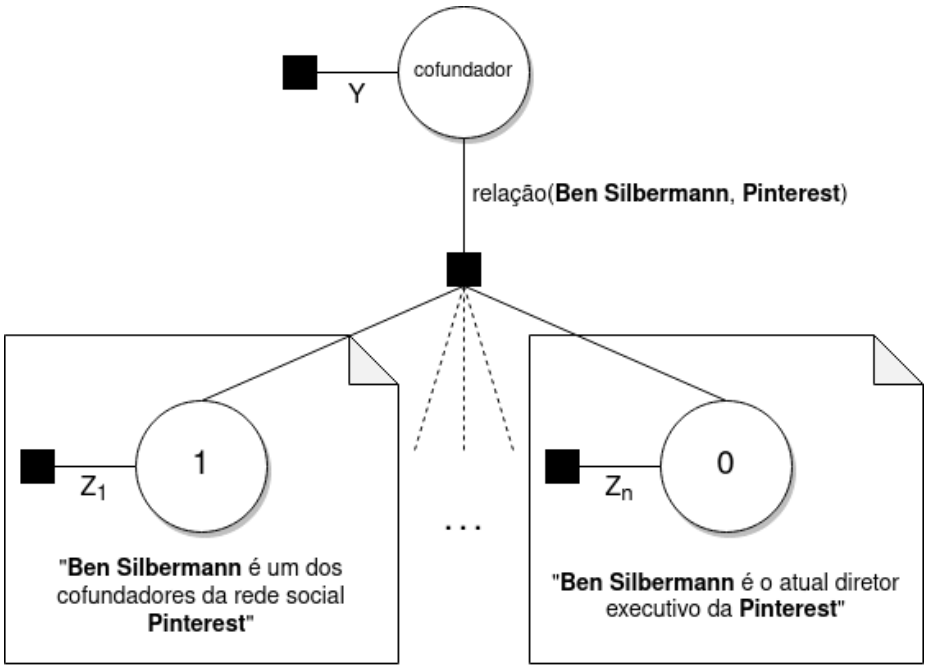

Figura 3.3: *Ilustração redesenhada do artigo de Riedel et al. (2010) da modelagem dos autores para extração de relações com supervisão distante utilizando um grafo de fatores. Círculos denotam variáveis e quadrados fatores (funções das variáveis adjacentes). A variável do topo denota o tipo da relação, enquanto que as variáveis associadas a cada sentença denotam se a sentença expressa ou não a relação.*

A modelagem apresentada é uma abordagem de múltipla instância com rótulo único. Utilizando novamente o exemplo da Figura 3.1, caso estivéssemos considerando tanto relações do tipo *cofundador* como relações do tipo *diretor executivo*, o modelo seria incapaz de inferir ambas as relações entre as entidades, e teria que decidir entre uma delas de acordo com o que aprendeu das sentenças utilizadas. Visando expandir essa abordagem, Hoffmann *et al.* (2011) e Surdeanu, Tibshirani *et al.* (2012) elaboraram modelos para o caso de múltipla instância com múltiplos rótulos. Citaremos brevemente as ideias por trás da modelagem de cada um dos autores, e assim como para o modelo de Riedel *et al.* (2010), recomenda-se aos leitores interessado nos detalhes internos por trás do funcionamento de cada modelo a leitura dos respectivos artigos.

O modelo de Hoffmann *et al.* (2011) ainda é uma abordagem baseada em um grafo de fatores, e a Figura 3.4 apresenta uma ilustração desse modelo. Para cada par de entidades e_1 e e_2, temos Y^r variáveis binárias para cada relação $r \in R$ do conjunto de relações predefinidas R. Similar ao modelo anterior, as variáveis Z_i estão associadas a cada uma das n sentenças nas quais e_1 e e_2 aparecem conjuntamente, mas não são mais binárias, podendo assumir quaisquer dos valores de R, além de um valor adicional que denote a ausência de um relacionamento (no exemplo da Figura 3.4, o valor "nenhum"). Aqui, temos dois tipos de fatores: fatores que só estão associados às variáveis Z_i e fatores associados a ambos os tipos de variáveis. Os fatores associados somente às variáveis Z_i tomam valores que dependem do valor atual de Z_i e de *features* extraídas da sentença associada. Diferente das abordagens anteriormente mostradas, as *features* utilizadas pelos autores são em nível de

sentença, e não em nível de *bag*, fazendo com que as extrações sejam fortemente guiadas pelas sentenças isoladas, e não por todo o conjunto de sentenças. Os fatores do outro tipo são responsáveis por garantir que as relações preditas em nível de *bag* só sejam consideradas caso ao menos uma sentença do *bag* seja predita com a mesma relação.

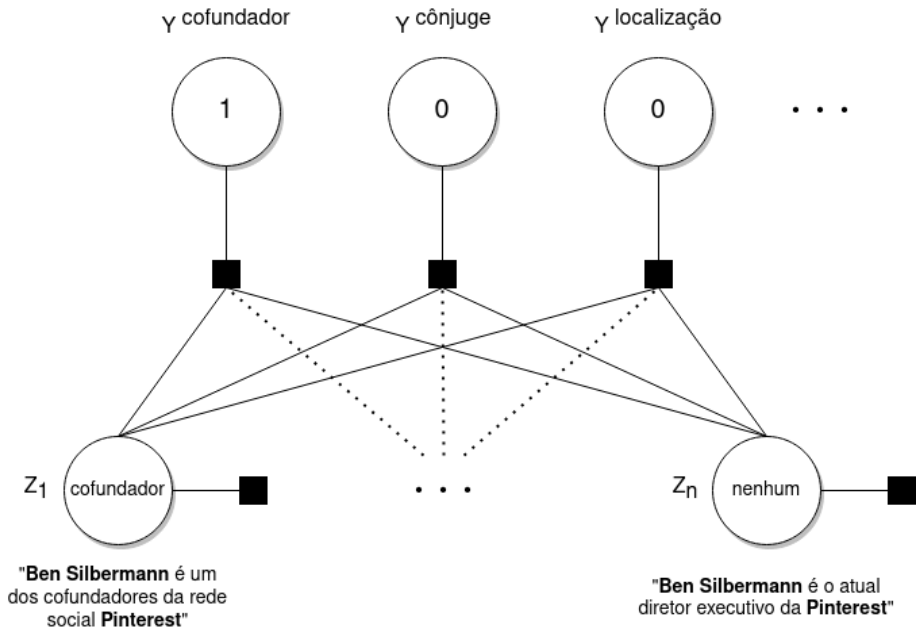

Figura 3.4: *Ilustração redesenhada do artigo de* Hoffmann *et al. (2011) da modelagem dos autores para múltipla instância com múltiplos rótulos. Círculos denotam variáveis e quadrados fatores (funções das variáveis adjacentes). A variáveis binárias Y^r denotam a expressão de cada relação r, enquanto que as variáveis associadas a cada sentença denotam a relação predita para a sentença associada.*

Na modelagem de Surdeanu, Tibshirani *et al.* (2012), temos duas camadas hierárquicas, e utiliza-se mais de um classificador para realizar as predições. Na camada mais interna, um classificador de múltiplas classes é responsável por atribuir rótulos a cada sentença individual. Na camada externa, um conjunto de classificadores binários, um para cada tipo de relação pré-definida, utilizam como entrada as predições de cada sentença para decidir se a relação associada a cada classificador se mantém entre as entidades. Segundo os autores, essa camada é importante na medida em que é responsável por capturar dependências entre as diferentes relações: por exemplo, o fato de que relações como *cônjuge* e *cofundador* são impossíveis para um mesmo par de entidades, ou que relações como *capital* e *localização* costumam ocorrer juntas. Para isso, os autores elaboraram *features* que fazem com que o modelo atribua pesos negativos ou positivos de acordo com as coocorrências de diferentes tipos de relações.

Na Figura 3.5, temos uma sobreposição das curvas de precisão-revocação dos modelos apresentados ao longo deste capítulo. Os resultados obtidos por todos os modelos discutidos nesta seção apontam para o fato de que a adoção de medidas que visam diminuir o impacto causado pelos ruídos da rotulação são benéficas. Além disso, mostram que permitir que um mesmo par de entidades exiba múltiplos rótulos também é benéfico para a extração. As curvas da Figura 3.5 foram obtidas a partir de experimentos de Surdeanu, Tibshirani *et al.* (2012). Neles, os autores compararam o desempenho de todos os modelos apresentados

neste capítulo no conjunto de dados criado por Riedel *et al.* (2010), e podemos notar que o modelo de Riedel *et al.* (2010) obteve valores de precisão consistentemente melhores do que o modelo de Mintz a partir de certo patamar baixo de revocação, enquanto que os modelos de Hoffmann *et al.* (2011) e Surdeanu, Tibshirani *et al.* (2012) obtiveram valores de precisão consideravelmente melhores do que os anteriores, para todos os níveis de revocação. No próximo capítulo, entraremos em detalhes nas primeiras modelagens baseadas na utilização de redes neurais, e que elaboraram modelos que se inspiraram nos resultados observados pelas abordagens discutidas neste capítulo.

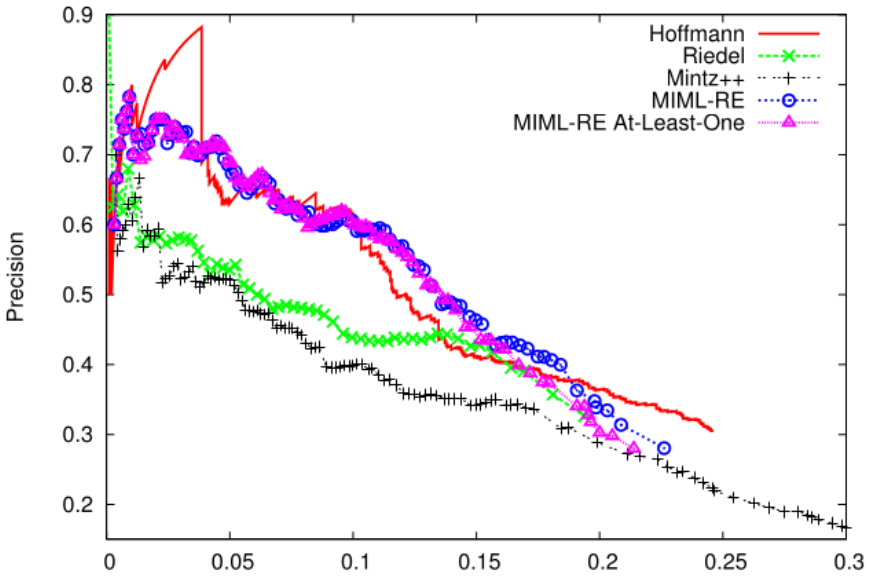

Figura 3.5: *Figura retirada do trabalho de Surdeanu, Tibshirani et al. (2012) com as curvas de precisão-revocação das abordagens de Mintz et al. (2009), Riedel et al. (2010), Hoffmann et al. (2011) e Surdeanu, Tibshirani et al. (2012) (MIML-RE), aplicadas ao conjunto de dados obtido por Riedel et al. (2010).*

Capítulo 4

Supervisão Distante via Redes Neurais

Atualmente, o estado da arte em extração de relações sob o paradigma de supervisão distante está na utilização de redes neurais. Abordagens como as de Nadgeri *et al.* (2021), Xu e Barbosa (2019) e Wu *et al.* (2019) são modeladas com base em redes neurais, e a avaliação desses modelos mostram resultados cada vez mais promissores. Com essa relevância em mente, nesse capítulo nos concentraremos em, de maneira semelhante ao capítulo anterior, ver um pouco da história por trás da utilização de redes neurais na tarefa de extração de relações. Neste processo, daremos destaque aos primeiros modelos desse tipo desenvolvidos, devido à sua relativa simplicidade e grande influência na área.

4.1 Contexto histórico

No Capítulo 3, apresentamos algumas das principais abordagens de extração de relações sob o paradigma de supervisão distante inicialmente desenvolvidas. Um ponto comum entre as abordagens apresentadas está no fato de que são abordagens baseadas em aprendizado de máquina estatístico, cujo desempenho depende fortemente das *features* utilizadas. Todas as abordagens apresentadas utilizaram *features* idênticas ou muito semelhantes às *features* utilizadas por Mintz *et al.* (2009), algumas das quais dependem do pré-processamento feito por meio de outras tarefas de processamento de linguagem, como a etiquetagem morfossintática e a construção da árvore de dependência das sentenças. Sendo também tarefas executadas por meio de aprendizado de máquina, seus resultados não são absolutamente precisos, e erros cometidos durante essa etapa de pré-processamento são propagados para a tarefa de extração de relações, prejudicando o desempenho nesta tarefa. Dessa forma, temos um cenário no qual o desempenho na tarefa de extração de relações depende das *features* utilizadas, e a qualidade das *features* depende das ferramentas de processamento de linguagem natural existentes.

Com o interesse em saber se a utilização de uma abordagem que não dependesse desse tipo de pré-processamento mais complicado era viável, Zeng, Liu, Lai *et al.* (2014) elaboraram um primeiro modelo de extração de relações baseado na utilização de redes neurais

convolucionais. A utilização de uma rede neural para isso surge como uma alternativa natural neste contexto, uma vez que é uma abordagem na qual as *features* são aprendidas de maneira automática durante a etapa de aprendizado, tirando a necessidade de que esforços sejam feitos para a elaboração delas. Além disso, a popularidade de redes neurais vinha se tornando cada vez maior nessa época, tanto em tarefas de processamento de linguagem natural como em demais áreas da computação. Certos trabalhos, como os de Sutskever (2013) e Mikolov *et al.* (2013), se destacavam por mostrarem como treinar determinadas estruturas de redes neurais de maneira eficiente. O trabalho de Mikolov *et al.* (2013), em especial, se destacou por introduzir maneiras eficientes de se treinar *word embeddings*, representações vetoriais de palavras (sobre as quais discutimos na Subseção 2.3.4) que se tornaram muito influentes na área de processamento de linguagem natural, ao se perceber que a inicialização de modelos com *word embeddings* previamente treinados aumentava o desempenho de modelos em diversas tarefas (Kim, 2014).

Este primeiro modelo de Zeng, Liu, Lai *et al.* (2014) foi elaborado para tentar realizar a tarefa de extração de relações fora do paradigma de supervisão distante, com um conjunto de dados muito menor do que os conjuntos de dados de supervisão distante usuais. Uma extensão desse modelo, visando adaptá-lo para o paradigma de supervisão distante, foi feita por Zeng, Liu, Y. Chen *et al.* (2015). Sendo uma extensão do primeiro modelo e que é mais relevante para nossos propósitos, descreveremos em detalhes apenas esse segundo modelo. As principais contribuições feitas por Zeng, Liu, Y. Chen *et al.* (2015) foram a utilização do que chamaram de rede neural convolucional por partes, visando capturar características de contexto das sentenças de forma mais precisa, e a adaptação do problema ao aprendizado de múltipla instância, visando diminuir a influência de erros causados pelos rótulos ruidosos do paradigma de supervisão distante. Como é de se esperar, a decisão dos autores de adaptar o problema ao aprendizado de múltipla instância foi inspirada pelo contexto de desenvolvimento da tarefa de extração de relações com supervisão distante, sobre o qual discutimos ao final da Seção 3.3, sendo uma forma de reduzir a influência negativa dos rótulos ruidosos no desempenho do modelo elaborado.

Por fim, falaremos ainda de mais uma extensão, dessa vez ao modelo de Zeng, Liu, Y. Chen *et al.* (2015). Essa extensão foi feita por Y. Lin *et al.* (2016), que propuseram a utilização de um mecanismo de atenção seletiva em nível de sentenças. Esse mecanismo é responsável por atribuir diferentes pesos às sentenças de um mesmo *bag*, sendo assim mais uma forma de tentar combater o problema causado por ruídos na rotulação com supervisão distante. Os autores optaram pela utilização desse mecanismo ao notarem que, embora Zeng, Liu, Y. Chen *et al.* (2015) tenham conseguido adaptar o problema ao aprendizado de múltipla instância, o fizeram de uma forma que nem toda a informação potencialmente importante de um *bag* de sentenças fosse considerada, uma vez que somente uma sentença de cada *bag* era utilizada.

4.2 Redes neurais convolucionais por partes

Na Figura 4.1, temos uma ilustração do que ocorre no modelo desenvolvido por Zeng, Liu, Y. Chen *et al.* (2015) em uma sentença específica. Nessa modelagem, o pré-processamento necessário é mínimo: só precisamos que a ligação de entidades seja feita, para que se saiba quais palavras representam as entidades nos textos fornecidos na entrada,

e para conseguirmos fazer o mapeamento dos resultados obtidos na saída para a base de conhecimento sendo utilizada. Fora isso, as *features* são aprendidas de forma automática pela rede neural convolucional utilizada, por meio de um processo melhor explicado na Subseção 2.2.2.

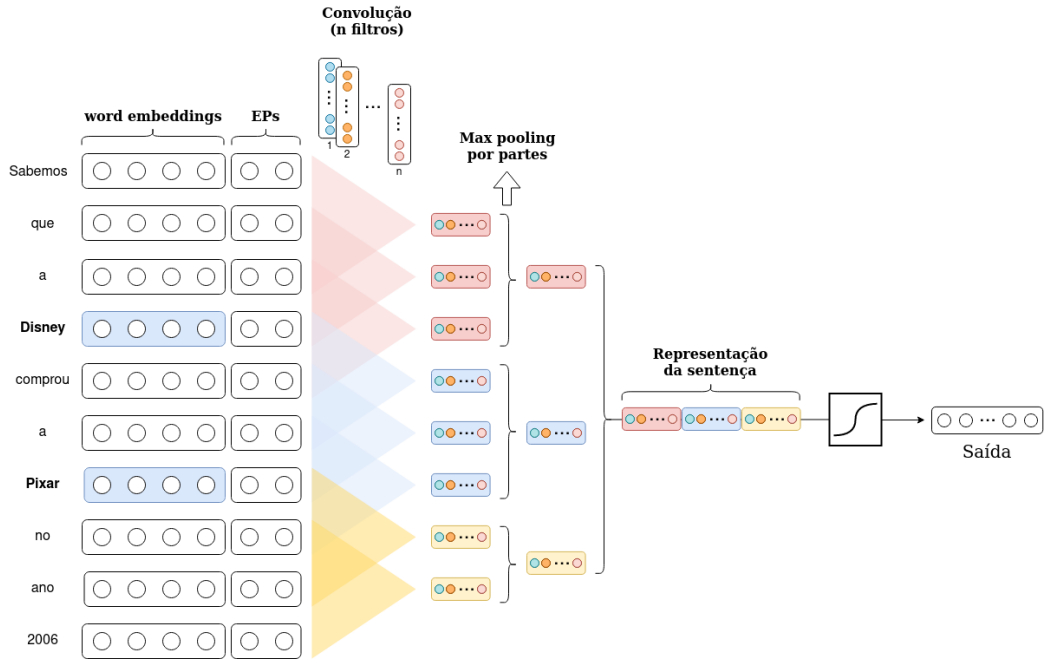

Figura 4.1: *Ilustração do modelo de redes neurais convolucionais por partes desenvolvido por Zeng, Liu, Y. Chen et al. (2015). Palavras são representadas pela concatenação de um word embedding com um embedding de posição (EP). A operação de pooling é feita por partes, em três regiões delimitadas pelas entidades da sentença. Cada valor da saída corresponde a uma das relações predefinidas utilizadas.*

O primeiro ponto a ser tratado é a representação vetorial das palavras utilizado. Como ilustrado na Figura 4.1, cada palavra é composta pela concatenação de dois tipos de vetores: *word embeddings* e *embeddings* de posição. As *word embeddings* utilizadas pelos autores, de tamanho d_w, foram obtidas por meio da utilização do modelo de *Skip-gram* apresentado por Mikolov *et al.* (2013), treinado no *corpus* utilizado — nesse trabalho, os autores se utilizaram do jornal *The New York Times* obtido por Riedel *et al.* (2010), mencionado na Seção 3.3. Os *embeddings* de posição foram propostos originalmente por Zeng, Liu, Lai *et al.* (2014), que os chamaram de *features* de posição. A ideia por trás desta proposta está em utilizar algo que sirva como uma maneira de codificar a posição de uma palavra em uma sentença. No trabalho original, uma *feature* de posição é definida como a combinação das distâncias relativas de uma palavra com cada uma das entidades e_1 e e_2 da sentença. Por exemplo, na Figura 4.1, temos a sentença "A *Disney* comprou a *Pixar* no ano 2006". Nela, a palavra *comprou* possui distâncias relativas 1 e -2, respectivamente, às entidades e_1 (*Disney*) e e_2 (*Pixar*). Para que esses valores sejam representados por meio de vetores, duas matrizes de *embeddings* de posição, $\mathbf{EP}_1$ e $\mathbf{EP}_2$, são aleatoriamente inicializadas, e cada distância relativa é transformada em um vetor de valores reais, de tamanho d_p, por meio da busca em cada uma das matrizes. A representação final é obtida concatenando o *word embedding* de cada palavra com os *embeddings* de posição de cada uma das distâncias relativas da palavra, resultando em um vetor de tamanho $d = d_w + 2d_p$. Por exemplo, na ilustração da

Figura 4.1, considera-se que $d = 6$, com $d_w = 4$ e $d_p = 1$. Na aplicação experimental do modelo, optaram empiricamente por utilizar dimensões $d_w = 50$ e $d_p = 5$, por esses valores apresentarem melhor desempenho.

A partir dessas representações, são feitas as convoluções. As convoluções ocorrem de maneira usual, assim como descrito na Subseção 2.2.2. Quanto aos parâmetros da convolução utilizados nos experimentos, utilizaram 230 filtros de convolução com janelas de tamanho 3. A principal diferença de uma rede neural convolucional usual para a abordagem utilizada pelos autores está na utilização de uma camada de *max pooling* por partes ao invés de uma camada de *max pooling* única, com o intuito de conseguir capturar informações de diferentes partes da sentença. Os autores dividem cada sentença em três segmentos, definidos em função das duas entidades — a primeira região contém as palavras à esquerda da primeira entidade, a segunda região contém as palavras entre as duas entidades, e a última região contém as palavras à direita da segunda. Para cada filtro de convolução, são extraídas as *features* de maior valor em relação a cada região, e assim, são extraídos três valores por filtro. Ao final, todos os valores de todos os filtros são concatenados em um único vetor, que pode ser entendido como a representação vetorial da sentença, com a importante propriedade que seu tamanho independe do tamanho da sentença. Na Figura 4.1, cada uma das regiões é representada pelas diferentes cores das janelas da convolução.

A decisão de utilizar *word embeddings* e *embeddings* de posição, assim como a utilização de *pooling* por partes, veio em função da literatura de extração de relações e processamento de linguagem natural existentes. Como previamente mencionado, trabalhos como o de Kim (2014) já haviam mostrado que a utilização de *word embeddings* previamente treinados aumentava o desempenho de tarefas de processamento de linguagem natural. A utilização de *embeddings* de posição e a separação do *pooling* em partes baseava-se nos trabalhos de extração de relação até então desenvolvidos, que costumavam utilizar *features* contendo informações relacionadas às posições das palavras. No próprio trabalho de Mintz *et al.* (2009), por exemplo, eram utilizadas *features* tais como as janelas de palavras adjacentes a cada entidade, as sequências de palavras entre as entidades, e o caminho de dependência entre as entidades. Na tentativa de capturar informações estruturais de maneira semelhante, os autores optaram pela elaboração e utilização dos mecanismos descritos, ao notarem que a utilização de *word embeddings* por si só era incapaz disso. A constatação de que esses componentes tiveram efeitos positivos sobre o desempenho do modelo foi feita de forma empírica. Para o *max pooling* em partes, Zeng, Liu, Y. Chen *et al.* (2015) compararam o desempenho entre a utilização de um *max pooling* global e singular com um *max pooling* em partes em uma rede neural convolucional. Na Figura 4.2, temos os resultados obtidos pelos autores, que apontam para a eficácia do *max pooling* em partes. Já em relação à utilização de *embeddings* de posição, sua eficácia foi evidenciada por Zeng, Liu, Lai *et al.* (2014), que compararam o desempenho do modelo com e sem a utilização desses tipos de *embeddings*, e notaram um ganho de 9.2% na estatística F do modelo (78.7% com contra 69.7% sem).

Obtida a representação vetorial da sentença por meio da concatenação dos vetores obtidos após a operação de *max pooling* em partes, aplica-se uma função de ativação e o vetor resultante é alimentado a um classificador para obter a saída. Seja r o número de relações predefinidas sendo consideradas, a saída é um vetor $\mathbf{o} = \{o_1, o_2, \ldots, o_r\}$ de valores

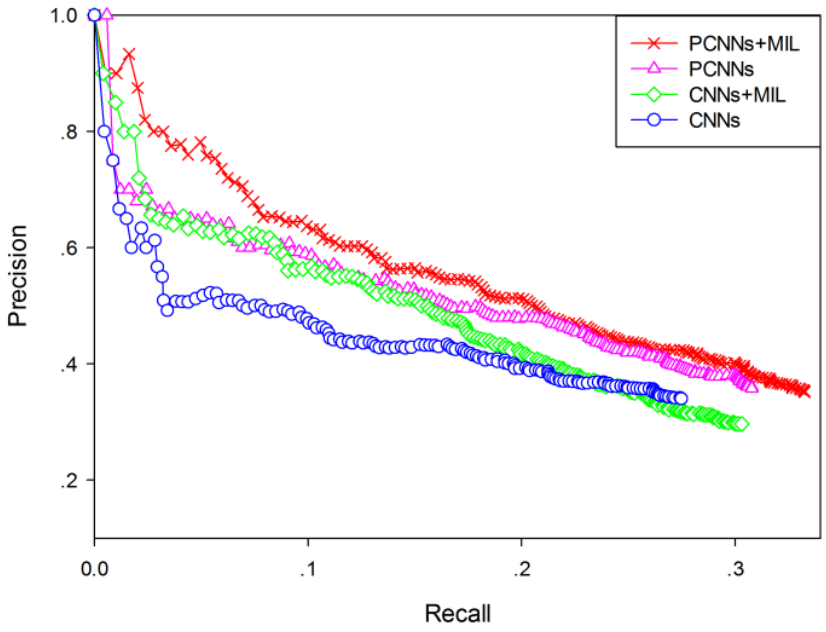

Figura 4.2: *Curva de precisão-revocação retirada do trabalho de Zeng, Liu, Y. Chen et al. (2015) comparando o desempenho de redes neurais convolucionais usuais (CNN) com redes neurais convolucionais por partes (PCNN), com e sem aprendizado de múltipla instância (MIL)*

reais, onde cada valor o_k corresponde ao valor de confiança da classificação da k-ésima relação. Cada um desses valores de confiança pode ser convertido em uma probabilidade condicional, e isso é um componente importante das mudanças adotadas pelos autores para adaptar essa modelagem do problema ao aprendizado de múltipla instância.

Como no aprendizado de múltipla instância são classificados *bags* ao invés de instâncias individuais, os autores precisaram utilizar uma função objetivo definida em função dos *bags*. A função objetivo elaborada foi definida utilizando a entropia cruzada em nível de *bags*. Primeiro, falaremos sobre como converter os valores de saída do vetor **o** em probabilidades condicionais. Considerando que todos os parâmetros ajustáveis do modelo são dados por θ, e que a saída **o** é obtida a partir de uma sentença qualquer s^*, podemos aplicar uma operação de *softmax* sobre todos os tipos de relações para obter a probabilidade condicional de uma relação k específica:

$$p(k|s^*;\theta) = \frac{e^{o_k}}{\sum_{i=1}^{r} e^{o_i}}$$

Com isso, podemos falar sobre a função objetivo utilizada. Considere que o conjunto de treinamento é dado por T pares (B_i, y_i), onde $B_i = \{s_i^1, s_i^2, \ldots, s_i^{n_i}\}$ corresponde ao i-ésimo *bag* de n_i sentenças s_i, e y_i o rótulo do i-ésimo *bag*, a função objetivo é dada por:

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i|s_i^{j^*};\theta)$$

Onde j^* é tal que:

$$j^* = \arg\max_j p(y_i|s_i^j;\theta), 1 \leq j \leq n_i$$

Ou seja, j^* corresponde ao índice da sentença do *bag* que maximiza a probabilidade condicional do valor de rotulação correta. Por meio dessa formulação, ao maximizar $J(\theta)$ com um algoritmo de retropropagação, as mudanças na rede neural são feitas de acordo com cada *bag* ao invés de serem feitas de acordo com cada instância. Com isso, a formulação utilizada pelos autores é capaz de incorporar a heurística de supervisão distante proposta por Riedel *et al.* (2010), e uma vez treinado o modelo, uma predição só atribui um rótulo positivo a um *bag* se a saída da rede neural atribuir o rótulo positivo a ao menos uma de suas sentenças.

Assim como foi feito para avaliar a eficácia da camada de *max pooling* em partes, os autores verificaram empiricamente que a utilização do aprendizado de múltipla instância aumentou o desempenho do modelo. Os resultados podem ser vistos na Figura 4.2.

Um problema mais evidente dessa adaptação está no fato de que nem toda a informação presente em um *bag* é utilizada. Como vimos pela definição de j^*, somente a sentença mais provável dentro de cada *bag* é escolhida, fazendo com que toda a informação contida nas sentenças ignoradas seja perdida. Sabendo disso, Y. Lin *et al.* (2016) propuseram a utilização de um mecanismo de atenção seletiva em nível de sentenças. Trata-se de um mecanismo capaz de atribuir diferentes pesos às sentenças de um *bag*, e o esperado é que essa atribuição ocorra de forma a minimizar a influência de rotulações ruidosas e maximizar a influência de rotulações corretas. Na Figura 4.3, temos uma ilustração do funcionamento desse mecanismo.

Considere que escolhemos um *bag* de sentenças $B = \{s_1, s_2, \ldots, s_n\}$ e seu respectivo rótulo y do conjunto de treinamento. Ao passar cada uma das s_i sentenças (com $1 \leq i \leq n$) pela rede neural convolucional em partes até agora discutida, obtemos como resultado um conjunto $C = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ em que cada $\mathbf{s}_i$ corresponde à representação vetorial da sentença obtida pela rede neural. Com o novo conjunto C em mãos, podemos calcular uma representação vetorial do *bag* como um todo por meio da soma ponderada de cada $\mathbf{s}_i$:

$$\mathbf{b} = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i$$

Para que esse mecanismo de atenção seletiva cumpra bem os seus propósitos, os pesos devem ser elaborados de forma que sejam capazes de diminuir o impacto das instâncias contendo rotulações com ruído. Os autores definem os pesos de forma que dependem de funções v_i, que dão um valor à compatibilidade da sentença com a relação:

$$\alpha_i = \frac{e^{v_i}}{\sum_{k=1}^{n} e^{v_k}}$$
$$v_i = \mathbf{s_i A r_i}$$

Aqui, $\mathbf{A}$ corresponde a uma matriz diagonal ponderada e $\mathbf{r}$ é o *embedding* de relação

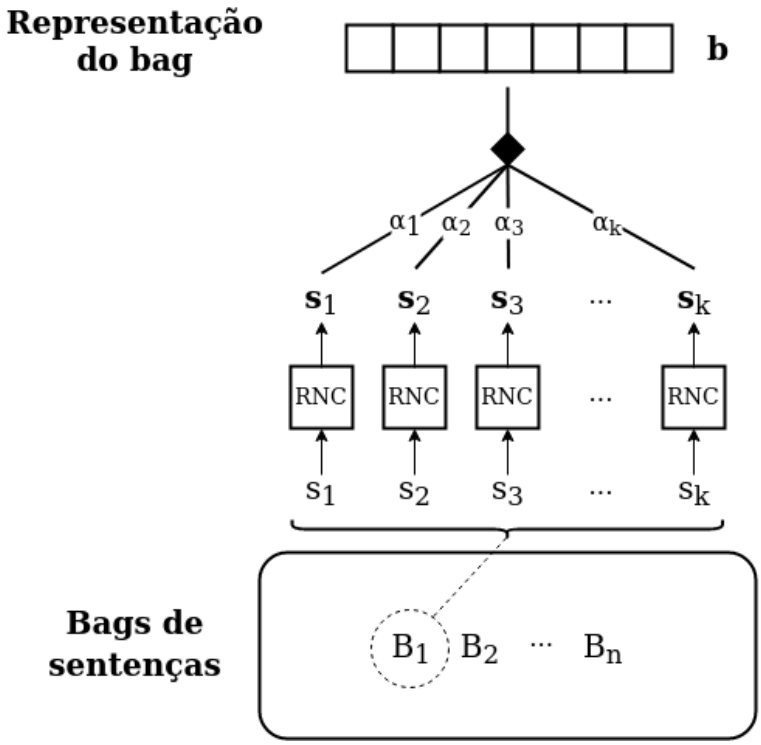

Figura 4.3: *Ilustração do funcionamento do mecanismo de atenção seletiva em nível de sentenças. De cada sentença s_i de um bag em particular, obtemos sua representação vetorial* $\mathbf{s}_i$ *por meio de uma rede neural convolucional (RNC). O mecanismo de atenção seletiva atribui um peso α_i a cada sentença, obtendo ao final uma representação* **b** *do bag de sentenças*

correspondente ao rótulo y, *embedding* esse obtido de uma matriz de *embeddings* das relações **R**. Os *embeddings* dessa matriz devem possuir o mesmo tamanho das representações das sentenças obtidas pela rede neural convolucional, para que os cálculos possam ser feitos da maneira adequada. A saída final da rede neural passa a ser calculada com base em **R**, da seguinte forma:

$$\mathbf{o} = \mathbf{Rb} + \mathbf{v}, \text{ onde } \mathbf{v} \text{ é um vetor de viés}$$

Além disso, podemos agora utilizar uma probabilidade condicional e uma função objetivo definidas agora em função dos *bags*:

$$p(k|B;\theta) = \frac{e^{o_k}}{\sum_{i=1}^{r} e^{o_i}}$$

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i|B_i;\theta)$$

Para avaliar a eficiência do mecanismo de atenção seletiva adotado, os autores novamente utilizaram o conjunto de dados obtido por Riedel *et al.* (2010) e realizaram experimentos comparando o desempenho da rede neural convolucional por partes antes e após a adoção desse mecanismo. Para isso, utilizaram diferentes formulações dos pesos α_i para obter modelos que serviram como patamar de comparação. A primeira dessas formu-

lações foi utilizar pesos fixos, atribuindo a mesma importância para todas as sentenças dos *bags*, ou seja, $\alpha_i = \frac{1}{n}$, $\forall i$. A segunda dessas foi incorporar o que foi feito no trabalho de Zeng, Liu, Y. Chen *et al.* (2015), em que somente a sentença mais provável era considerada. Isso pode ser interpretado como um caso especial de atenção seletiva, em que $\alpha_i = 1$ para a sentença mais provável e $\alpha_i = 0$ para as demais. Na Figura 4.4, temos as curvas de precisão-revocação de cada um desses modelos sobrepostas. Na figura, PCNN+ONE corresponde ao modelo com um mecanismo de atenção seletiva que só escolhe a sentença mais provável, PCNN+AVE corresponde ao modelo com o mecanismo de atenção seletiva que utiliza o peso médio, PCNN+ATT corresponde ao modelo com o mecanismo de atenção seletiva que utiliza a matriz diagonal ponderada e, por fim, PCNN corresponde à modelagem original proposta por Zeng, Liu, Y. Chen *et al.* (2015), mas sem a adaptação do modelo ao aprendizado de múltipla instância. Como é possível observar, o modelo com o mecanismo de atenção seletiva elaborado pelos autores teve um desempenho consistentemente melhor aos outros, em todos os níveis de revocação.

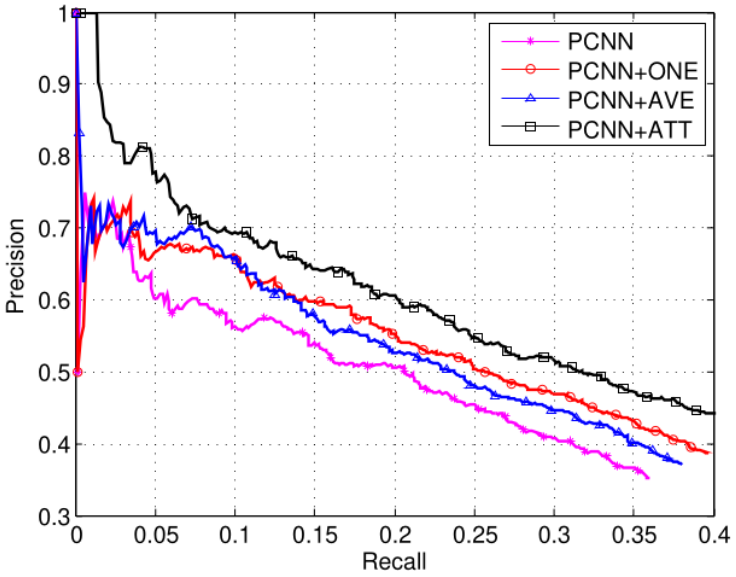

Figura 4.4: *Gráfico retirado do trabalho de Y. Lin et al. (2016) comparando, por meio das curvas de precisão-revocação, o desempenho de redes neurais convolucionais por partes utilizando diferentes mecanismos de atenção seletiva em nível de sentenças.*

Por fim, é interessante compararmos o desempenho das abordagens de redes neurais convolucionais em partes vistas nesse capítulo com as abordagens apresentadas no Capítulo 3. Na Figura 4.5, apresentamos as curvas de precisão-revocação de algumas das diferentes abordagens discutidas ao longo deste trabalho. Na figura, as abordagens são nomeadas da seguinte forma:

- *Mintz*: corresponde ao primeiro modelo de supervisão distante, de Mintz *et al.* (2009)
- *MultiR*: corresponde ao modelo de supervisão distante com aprendizado de múltipla instância de Hoffmann *et al.* (2011)
- *MIML*: corresponde ao modelo de supervisão distante com aprendizado de múltipla instância com múltiplos rótulos de Surdeanu, Tibshirani *et al.* (2012)

- *CNN+ATT*: corresponde ao modelo de redes neurais convolucionais, sem a utilização do *max pooling* em partes, com o mecanismo de atenção seletiva, elaborado por Y. Lin *et al.* (2016)
- *PCNN+ATT*: idêntico ao anterior, mas com a utilização do *max pooling* em partes, como proposto por Zeng, Liu, Y. Chen *et al.* (2015)

Como podemos ver na figura, as modelagens baseadas na utilização de redes neurais convolucionais conseguiram resultados claramente superiores às abordagens anteriores. Isso é ainda mais evidente para níveis mais elevados de revocação — na curva, podemos ver uma queda mais acentuada da precisão nos modelos anteriores a partir do nível de revocação de 10%, enquanto que as abordagens de redes neurais convolucionais apresentam uma queda mais suave. É importante notar que, apesar disso, os níveis de precisão a partir da revocação de 20% já são consideravelmente baixos, mesmo para as abordagens de redes neurais.

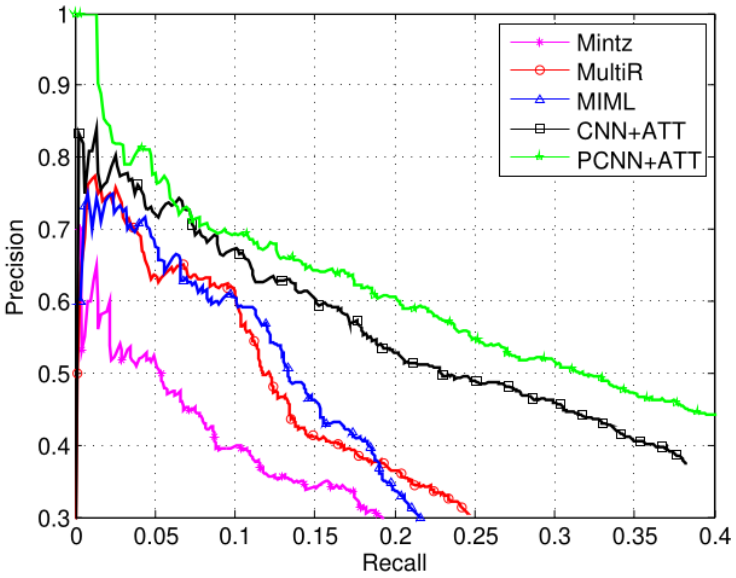

Figura 4.5: *Gráfico retirado do trabalho de Y. Lin et al. (2016) comparando, por meio das curvas de precisão-revocação, o desempenho de diversos modelos de extração de relações.*

Capítulo 5

Conclusão

Neste trabalho, nos propusemos a elaborar um texto introdutório à tarefa de extração de relações sob o paradigma de supervisão distante. Para isso, apresentamos ao longo do texto algumas das primeiras abordagens desenvolvidas na literatura, tanto por serem altamente influentes como por serem relativamente simples, facilitando o entendimento de alguns dos principais conceitos e decisões de modelagem que podem ser levadas em consideração ao tentar realizar a tarefa. A avaliação dos trabalhos apresentados mostra que o paradigma de supervisão distante mostra-se como uma alternativa viável a outros tipos de abordagens de extração de relações da literatura, obtendo boa precisão sob certas condições. Independente da eficácia e viabilidade dessas abordagens, espera-se que este texto sirva como uma boa introdução a alguns dos caminhos que podem ser tomados e algumas decisões que merecem destaque ao realizar a extração de relações sob esse paradigma.

As abordagens descritas com mais detalhes são relativamente simples, principalmente comparando-as a outras abordagens de extração de relações existentes na bibliografia. Os modelos baseados em redes neurais se destacam nesse sentido, podendo ser considerados ainda mais simples uma vez que dispensam a elaboração de *features*, além de se utilizarem de estruturas de redes neurais bem conhecidas. Apesar dessas observações, a aplicação de fato das abordagens mostradas pode se mostrar uma tarefa mais complicada do que aparenta ser. O fato de que a supervisão distante necessita de uma base de conhecimento para sua aplicação acaba limitando consideravelmente o escopo de aplicação dessa abordagem. Por mais que existam hoje bases de conhecimentos de larga escala disponíveis de forma aberta, costumam ser bases de conhecimento que armazenam conhecimentos mais gerais, fazendo que sua aplicação para a extração de relações mais específicas possa não funcionar tão bem. Podemos nos encontrar em uma situação em que as relações pertinentes não existem na base de conhecimento, ou que até existam mas sejam escassas, contendo um número pequeno de instâncias, que podem ser insuficientes para que a extração das relações no *corpus* utilizado consiga capturar os padrões necessários.

Além disso, as avaliações feitas pelos autores dos modelos estudados deixam algumas dúvidas. O primeiro ponto que podemos levantar, relacionado ao que foi dito no parágrafo anterior, está na aplicabilidade dessas técnicas para certos domínios mais específicos. O conjunto de dados desenvolvido por Riedel *et al.* (2010) se tornou o principal conjunto

de dados para avaliar o desempenho dessas modelagens, mas foi elaborado utilizando relações da *Freebase* de somente quatro categorias mais gerais (*people, business, person, location*). Não sabemos como essas abordagens se comportariam se aplicadas a *corpus* de texto e relações de um domínio mais específico. Além disso, seria interessante que fossem mostradas mais informações acerca da avaliação. Seria útil, por exemplo, termos dados não somente acerca da precisão global, mas também acerca da precisão local, referente a cada um dos tipos de relações sendo consideradas, para termos uma visão mais completa acerca do desempenho dos modelos. Não sabemos, por exemplo, se os altos valores de precisão obtidos fazem referência a apenas poucos tipos de relações que ocorrem com muito mais frequência nos dados, ou se os modelos possuem desempenho até melhor do que o observado para um conjunto específico de relações.

Embora o tempo reduzido aliado à extensa bibliografia estudada tenham sido os principais obstáculos encontrados por nós, dificuldades como a apontada anteriormente, em relação às bases de conhecimento, contribuíram em parte para que, neste trabalho, não tenhamos conseguido desenvolver nenhuma aplicação de fato, embora esse fosse inicialmente um de nossos objetivos. Uma possibilidade de estudo futura está na aplicação dos modelos observados para dados de outras línguas, como o português, buscando avaliar a efetividade e capacidade de generalização destes modelos para além do inglês. Imagina-se que tal estudo seja viável, uma vez que é possível encontrar *corpus* de textos de outras línguas disponíveis de forma aberta, e que existem bases de conhecimento como o Wikidata, que aparentam possuir grande volume de informações em diversas línguas. Para esse fim, a utilização do *OpenNRE* (Han *et al.*, 2020), um *toolkit* de *Python* feito especificamente para a implementação de modelos de extração de relações baseados em redes neurais, parece ser uma boa escolha com base no contato preliminar que tivemos com essa ferramenta ao longo do desenvolvimento deste trabalho. Outra alternativa de trabalho futuro está em estudar abordagens diferentes dessa área. Seriam interessantes estudos de abordagens baseadas em outras estruturas de redes neurais, tais como redes neurais recursivas ou redes neurais recorrentes. Uma outra possibilidade de tema está em estudar a extração de relações em nível de documento, na qual o contexto de um documento como um todo é considerado para a extração. Essa é uma configuração do problema especialmente interessante, não somente devido a sua maior complexidade mas também devido ao seu grande potencial, uma vez que determinadas relações só podem ser extraídas se considerarmos contextos distantes dentro de um mesmo documento.

Referências

[Abu-Mostafa *et al.* 2012] Yaser S. Abu-Mostafa, Malik. Magdon-Ismail e H T Lin. "Learning from Data: A Short Course". Em: *AMLBook*. 2012 (citado na pg. 6).

[Brin 1999] Sergey Brin. "Extracting patterns and relations from the world wide web". Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1590. 1999. DOI: 10.1007/10704656_11 (citado na pg. 22).

[P. P. S. Chen 1976] Peter Pin Shan Chen. "The Entity-Relationship Model—toward a Unified View of Data". Em: *ACM Transactions on Database Systems (TODS)* 1.1 (1976). ISSN: 15574644. DOI: 10.1145/320434.320440 (citado na pg. 3).

[Giesbrecht e Evert 2009] Eugenie Giesbrecht e Stefan Evert. "Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus". Em: *Web as Corpus Workshop WAC5* (2009) (citado na pg. 16).

[Han *et al.* 2020] Xu Han *et al.* "OpenNRE: An open and extensible toolkit for neural relation extraction". Em: *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Proceedings of System Demonstrations*. 2020. DOI: 10.18653/v1/d19-3029 (citado nas pgs. 4, 44).

[Hoffmann *et al.* 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer e Daniel S. Weld. "Knowledge-based weak supervision for information extraction of overlapping relations". Em: *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. 2011 (citado nas pgs. 30–32, 40).

[Kim 2014] Yoon Kim. "Convolutional neural networks for sentence classification". Em: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. 2014. DOI: 10.3115/v1/d14-1181 (citado nas pgs. 34, 36).

[Lehmann *et al.* 2015] Jens Lehmann *et al.* "DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia". Em: *Semantic Web* 6.2 (2015). ISSN: 22104968. DOI: 10.3233/SW-140134 (citado na pg. 19).

[D. Lin 1998] Dekang Lin. "Automatic retrieval and clustering of similar words". Em: 1998. DOI: 10.3115/980691.980696 (citado na pg. 17).

[Y. Lin *et al.* 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan e Maosong Sun. "Neural relation extraction with selective attention over instances". Em: *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*. Vol. 4. 2016. DOI: 10.18653/v1/p16-1200 (citado nas pgs. 2, 34, 38, 40, 41).

[Manning 2011] Christopher D. Manning. "Part-of-speech tagging from 97% to 100%: Is it time for some linguistics?" Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 6608 LNCS. 2011. DOI: 10.1007/978-3-642-19400-9_14 (citado na pg. 16).

[Mikolov *et al.* 2013] Tomas Mikolov, Kai Chen, Greg Corrado e Jeffrey Dean. "Efficient estimation of word representations in vector space". Em: *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. 2013 (citado nas pgs. 17, 18, 34, 35).

[Mintz *et al.* 2009] Mike Mintz, Steven Bills, Rion Snow e Dan Jurafsky. "Distant supervision for relation extraction without labeled data". Em: 2009. DOI: 10.3115/1690219.1690287 (citado nas pgs. 1, 2, 21–23, 26–29, 32, 33, 36, 40).

[Nadgeri *et al.* 2021] Abhishek Nadgeri *et al.* "KGPool: Dynamic Knowledge Graph Context Selection for Relation Extraction". Em: 2021. DOI: 10.18653/v1/2021.findings-acl.48 (citado na pg. 33).

[Pennington *et al.* 2014] Jeffrey Pennington, Richard Socher e Christopher D. Manning. "GloVe: Global vectors for word representation". Em: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. 2014. DOI: 10.3115/v1/d14-1162 (citado na pg. 18).

[Riedel *et al.* 2010] Sebastian Riedel, Limin Yao e Andrew McCallum. "Modeling relations and their mentions without labeled text". Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 6323 LNAI. PART 3. 2010. DOI: 10.1007/978-3-642-15939-8_10 (citado nas pgs. 28, 30, 32, 35, 38, 39, 43).

[Rosenfeld e Feldman 2007] Benjamin Rosenfeld e Ronen Feldman. "Using corpus statistics on entities to improve semi-supervised relation extraction from the Web". Em: *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. 2007 (citado na pg. 22).

[Russell e Norvig 2010] Stuart Russell e Peter Norvig. *Artificial Intelligence A Modern Approach Third Edition*. 2010. DOI: 10.1017/S0269888900007724 (citado nas pgs. 5, 6).

[Shinyama e Sekine 2006] Yusuke Shinyama e Satoshi Sekine. "Preemptive Information Extraction using Unrestricted Relation Discovery". Em: *HLT-NAACL 2006 - Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings of the Main Conference*. 2006. DOI: 10.3115/1220835.1220874 (citado na pg. 22).

[Smirnova e Cudré-Mauroux 2019] Alisa Smirnova e Philippe Cudré-Mauroux. *Relation extraction using distant supervision: A survey*. 2019. DOI: 10.1145/3241741 (citado na pg. 19).

[Suchanek *et al.* 2007] Fabian M. Suchanek, Gjergji Kasneci e Gerhard Weikum. "Yago: A core of semantic knowledge". Em: *16th International World Wide Web Conference, WWW2007*. 2007. DOI: 10.1145/1242572.1242667 (citado na pg. 19).

[Surdeanu e Ciaramita 2007] Mihai Surdeanu e Massimiliano Ciaramita. "Robust Information Extraction with Perceptrons". Em: *Proceedings of the NIST 2007 Automatic Content Extraction Workshop ACE07* (2007) (citado nas pgs. 21, 27).

[Surdeanu, Tibshirani *et al.* 2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati e Christopher D. Manning. "Multi-instance multi-label learning for relation extraction". Em: *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*. 2012 (citado nas pgs. 30–32, 40).

[Sutskever 2013] Ilya Sutskever. "Training Recurrent neural Networks". Tese de dout. University of Toronto, 2013 (citado na pg. 34).

[Vrandečić 2012] Denny Vrandečić. "Wikidata: A new platform for collaborative data collection". Em: *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*. 2012. DOI: 10.1145/2187980.2188242 (citado na pg. 19).

[Wu *et al.* 2019] Shanchan Wu, Kai Fan e Qiong Zhang. "Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector". Em: *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*. 2019. DOI: 10.1609/aaai.v33i01.33017273 (citado nas pgs. 2, 33).

[Xu e Barbosa 2019] Peng Xu e Denilson Barbosa. "Connecting language and knowledge with heterogeneous representations for neural relation extraction". Em: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. Vol. 1. 2019. DOI: 10.18653/v1/n19-1323 (citado nas pgs. 2, 33).

[Zeng, Liu, Y. Chen *et al.* 2015] Daojian Zeng, Kang Liu, Yubo Chen e Jun Zhao. "Distant supervision for relation extraction via Piecewise Convolutional Neural Networks". Em: *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*. 2015. DOI: 10.18653/v1/d15-1203 (citado nas pgs. 2, 34–37, 40, 41).

[Zeng, Liu, Lai *et al.* 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou e Jun Zhao. "Relation classification via convolutional deep neural network". Em: *COLING 2014 - 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*. 2014 (citado nas pgs. 33–36).

[G. D. Zhou *et al.* 2005] Guo Dong Zhou, Jian Su, Jie Zhang e Min Zhang. "Exploring various knowledge in relation extraction". Em: *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. 2005. DOI: 10.3115/1219840.1219893 (citado na pg. 21).

[P. Zhou *et al.* 2016] Peng Zhou *et al.* "Attention-based bidirectional long short-term memory networks for relation classification". Em: *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*. 2016. DOI: 10.18653/v1/p16-2034 (citado na pg. 2).

[Zhu *et al.* 2020] Hao Zhu *et al.* "Graph neural networks with generated parameters for relation extraction". Em: *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. 2020. DOI: 10.18653/v1/p19-1128 (citado na pg. 2).

Índice Remissivo