Universidade de São Paulo
Faculdade de Filosofia, Letras e Ciências Humanas
Departamento de Linguística
Programa de Pós-Graduação em Semiótica e Linguística Geral

Comparação de Métodos para Inferência em Linguagem Natural

[Versão Corrigida]

Rodrigo Aparecido da Silva Souza

Dissertação apresentada ao Programa de Pós-Graduação em Semiótica e Linguística Geral do Departamento de Linguística da Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo para obtenção do título de Mestre em Letras.

Orientador: Prof. Dr. Marcos Lopes

São Paulo
2020

Universidade de São Paulo
Faculdade de Filosofia, Letras e Ciências Humanas
Departamento de Linguística
Programa de Pós-Graduação em Semiótica e Linguística Geral

Comparação de Métodos para Inferência em Linguagem Natural

[Versão Corrigida]

Rodrigo Aparecido da Silva Souza

São Paulo
2020

Catalogação na Publicação
Serviço de Biblioteca e Documentação
Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo

S719c

Souza, Rodrigo Aparecido da Silva
Comparação de Métodos para Inferência em Linguagem Natural / Rodrigo Aparecido da Silva Souza; orientador Marcos Fernando Lopes - São Paulo, 2020.
124 f.

Dissertação (Mestrado)- Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo. Departamento de Linguística. Área de concentração: Semiótica e Lingüística Geral.

1. Linguística Computacional. 2. Inferência em Linguagem Natural . 3. Bag-of-Words. 4. Alinhamento. 5. BERT. I. Lopes, Marcos Fernando , orient. II. Título.

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, LETRAS E CIÊNCIAS HUMANAS

ENTREGA DO EXEMPLAR CORRIGIDO DA DISSERTAÇÃO/TESE

Termo de Ciência e Concordância do (a) orientador (a)

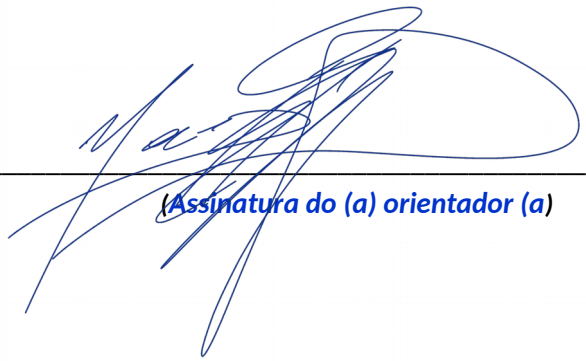
Nome do (a) aluno (a): Rodrigo Aparecido da Silva Souza

Data da defesa: 18/12/2020

Nome do Prof. (a) orientador (a): Marcos Fernando Lopes

Nos termos da legislação vigente, declaro ESTAR CIENTE do conteúdo deste **EXEMPLAR CORRIGIDO** elaborado em atenção às sugestões dos membros da comissão Julgadora na sessão de defesa do trabalho, manifestando-me **plenamente favorável** ao seu encaminhamento e publicação no **Portal Digital de Teses da USP**.

São Paulo, 17/2/21.



_____

(Assinatura do (a) orientador (a)

A presente pesquisa foi realizada com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo no 133781/2019-4.

This research was financed in part by The Brazilian National Council for Scientific and Technological Development (CNPq), grant 133781/2019-4.

Rio Sem Discurso

Quando um rio corta, corta-se de vez
o discurso-rio de água que ele fazia;
cortado, a água se quebra em pedaços,
em poços de água, em água paralítica.
Em situação de poço, a água equivale
a uma palavra em situação dicionária:
isolada, estanque no poço dela mesma,
e porque assim estanque, estancada;
e mais: porque assim estancada, muda,
e muda porque com nenhuma comunica,
porque cortou-se a sintaxe desse rio,
o fio de água por que ele discorria.

O curso de um rio, seu discurso-rio,
chega raramente a se reatar de vez;
um rio precisa de muito fio de água
para refazer o fio antigo que o fez.
Salvo a grandiloquência de uma cheia
lhe impondo interina outra linguagem,
um rio precisa de muita água em fios
para que todos os poços se enfrasem:
se reatando, de um para outro poço,
em frases curtas, então frase e frase,
até a sentença-rio do discurso único
em que se tem voz a seca ele combate.

João Cabral de Melo Neto

Aos meus pais: Pedro e Fátima

Agradecimentos

Ao meu orientador, Prof. Dr. Marcos Lopes, pelos anos de aprendizado e pela confiança depositada em mim desde a Iniciação Científica. Sou grato pelo apoio, pelas discussões sobre a linguística, sobre a vida e sobre a carreira acadêmica. Agradeço, também, pelo comprometimento enquanto orientador e por sua preocupação com minha formação acadêmica.

Aos Profs. Drs. do Departamento de Linguística da USP, pelas matérias ministradas com tanto rigor e por todo o conhecimento que me ofereceram desde os meus primeiros anos de USP. Agradeço, também, ao apoio técnico e à atenção dos funcionários do Departamento de Linguística.

Ao GLIC, Grupo de Estudos em Linguística Computacional, pelo acolhimento durante meus primeiros passos na Linguística Computacional e pelas discussões produtivas. A participação nos encontros do grupo e nos Workshops contribuíram substancialmente para minha formação e pesquisa.

Aos Profs. Drs. integrantes da banca de qualificação e defesa, Paulo Chagas de Souza, Marcelo Barra Ferreira, Marcello Modesto dos Santos, Edson Satoshi Gomi, Pablo Picasso Feliciano de Faria, Leonel Figueiredo de Alencar e Marcos Ribeiro Pereira Barretto. Agradeço a todos pelas sugestões e leitura atenta do meu trabalho. Ao professor Edson Satoshi Gomi, agradeço também por me receber tão bem e me auxiliar com dúvidas sobre o Weka.

Por fim, à minha família, a base humana sobre a qual esta pesquisa foi construída. Aos meus pais, Pedro e Fátima, pelo carinho, pela importância dada as minhas escolhas e por sempre priorizarem meus estudos. Às minhas irmãs, Rosana, minha segunda mãe, e Renata, pelo incentivo, apoio e carinho. Ao meu irmão, Renan, pelo apoio, pelas discussões sobre pesquisa, carreira e vida. Sou muito grato por cada momento que passamos juntos. Por fim, às minhas queridas sobrinhas, Isadora e Íris, que deram um novo sentido pra minha vida. Amo cada momento que passo ao lado de vocês.

Sumário

Lista de Tabelas

Lista de Abreviações

BoW	Bag-of-Words
BACBI	Bag-of-Words, Alinhamento e Classificador Bayesiano Ingênuo
BCBI	Bag-of-Words e Classificador Bayesiano Ingênuo
CD	Comparable Documents
IE	Information Extraction
IDF	Inverse Document Frequency
IR	Information Retrieval
MT	Machine Translation
NER	Named Entity Recognition
NLI	Natural Language Inference
NLP	Natural Language Processing
PLN	Processamento de Linguagem Natural
QA	Question Answering
PP	Paraphrase Acquisition
RC	Reading Comprehension
RTE	Recognizing Textual Entailment
SUM	Multi-document Summarization

Resumo

A Inferência em Linguagem Natural, do inglês *Natural Language Inference* (NLI), é um dos tópicos de pesquisa do Processamento Computacional de Linguagem Natural. Consiste, basicamente, na tarefa de determinar se um texto breve em língua natural, chamado *premissa*, acarreta outro texto, chamado *hipótese*. Normalmente, a tarefa é apresentada em forma de pares de premissa-hipótese e uma classificação para a relação de acarretamento.

Neste trabalho, propomo-nos a testar diferentes métodos de solução para os problemas de NLI oferecidos pelos três primeiros conjuntos de dados do *Pascal RTE Challenge* (Dagan *et al.*, 2005), o RTE-1, o RTE-2 e o RTE-3. Para tanto, implementamos quatro métodos diferentes de solução e algumas combinações entre eles: um método baseado em regras de Bag-of-Words (BoW) sem alinhamento, um baseado em alinhamento sentencial, um baseado em representação lógica para os textos dos pares e um baseado na tarefa de *Question Answering* (QA). Nosso objetivo é comparar em que medida métodos baseados em regras são eficazes para solucionar problemas de NLI e em que medida podem concorrer minimamente com modelos baseados em arquiteturas *Transformer* como o RoBERTa (Liu *et al.*, 2019*b*), cujo desempenho é reconhecidamente bom nessa tarefa.

A partir da implementação de diferentes regras de classificação, compusemos dois modelos. O primeiro, chamado BCBI, foi composto por regras de BoW sem alinhamento e por um Classificador Bayesiano Ingênuo. O segundo, chamado BACBI, foi composto regras de BoW, métodos de alinhamento e por um Classificador Bayesiano Ingênuo. O BCBI obteve uma acurácia de 65% no RTE-1, 57% no RTE-2 e 63% no RTE-3. O modelo BACBI obteve uma acurácia de 55% no RTE-1, 57% no RTE-2 e 60% no RTE-3.

Para o teste baseado em QA, convertemos hipóteses em perguntas polares (sim/não) e mantivemos as premissas como se fossem candidatas a respostas. As duas são passadas para o modelo RoBERTa para a classificação dos pares. Avaliado nos conjuntos de dados, o modelo atingiu uma acurácia de 74% no RTE-1, 78% no RTE-2 e 71% no RTE-3.

Por fim, comparamos os resultados alcançados pelos modelos com outros trabalhos avaliados nos conjuntos de dados. Concluímos que os modelos baseados em regras não foram eficazes para solucionar os problemas da tarefa. O método baseado no modelo RoBERTa, no entanto, atingiu resultados compatíveis com as melhores classificações nos corpora relatadas na literatura.

Palavras-Chave: Linguística Computacional, Inferência em Linguagem Natural, modelos Bag-of-Words, modelos com alinhamento, Representação Lógica, RoBERTa.

Abstract

Natural Language Inference (NLI) is a research topic in Natural Language Processing. In short, NLI is the task of determining if a natural language text, called *premise*, entails another text, the *hypothesis*. Generally, the problem is formulated as a premise-hypothesis pair together with a classification label for the entailment relation.

In this work, we propose to test different classification methods for the NLI problems offered by the first three datasets of the Pascal RTE Challenge benchmarks (Dagan *et al.*, 2005), RTE-1, RTE-2 and RTE-3. For this purpose, we implemented four different classification methods and some combinations between them: one based on Bag-of-Words (BoW) without alignment, one based on sentence alignment, one based on logical representation of the pairs, and one based on the Question Answering (QA) task. Our goal is to compare them, to evaluate the effectiveness of rule-based methods in solving the NLI problems, and to find out to what extent they could compete with Transformers based models such as RoBERTa (Liu *et al.*, 2019*b*) that are generally recognized for their performance in this task.

From the implementation of different classification rules, we composed two models. The first one, called BCBI, was composed by BoW rules without alignment, and a Naive Bayes Classifier. The second model, called BACBI, was composed by BoW rules, alignment methods, and a Naive Bayes Classifier. The BCBI model achieved an accuracy of 65% in the RTE-1, 57% in the RTE-2 and 63% in the RTE-3 dataset. The BACBI model, in turn, achieved an accuracy of 55% in the RTE-1, 57% in the RTE-2 and 60% in the RTE-3.

For the QA-based test, we converted the hypothesis into polar questions (yes/no) and used the premises as candidates for answers. Both were input in the RoBERTa model for the pairs classification. This model achieved an accuracy of 74% in the RTE-1, 78% in the RTE-2 and 71% in the RTE-3.

Finally, we compared the results achieved by the models with other works evaluated in the same datasets. The rule-based models were not as efficient as the Transformer based model in solving NLI tasks. The latter achieved results comparable to some of the best classifiers for the RTE datasets.

Keywords: Computational Linguistics, Natural Language Inference, Bag-of-Words, Alignment, Logical Representation, RoBERTa.

1

Introdução

1.1 A Inferência em Linguagem Natural

A Inferência em Linguagem Natural, do inglês *Natural Language Inference* (NLI), também chamada de Reconhecimento de Implicação Textual, em inglês *Recognizing Textual Entailment* (RTE), é um dos tópicos de pesquisa do Processamento de Linguagem Natural, ou *Natural Language Processing* (NLP). Consiste, basicamente, em determinar se um texto breve em língua natural, chamado premissa, acarreta outro texto, chamado hipótese. Desde a sua criação em 2005 com o *Pascal RTE Challenge*[1], a NLI tem ganhado cada vez mais espaço entre a comunidade científica de NLP.

Ao falarmos de sua criação, estamos nos referindo especificamente à NLI como a conhecemos atualmente: quase sempre uma tarefa compartilhada, na qual um conjunto de dados é disponibilizado para a avaliação dos mais variados tipos de metodologias e modelos. Antes do *Pascal RTE Challenge*, no entanto, a avaliação de acarretamentos entre textos já havia sido objeto de reflexão teórica e prática para alguns pesquisadores.

Uma proposta similar pode ser encontrada em Cooper *et al.* (1996), pesquisadores que apresentaram o CLEARS, *Computational Semantics Tool for Education and Research System*, e o corpus FraCas. O CLEARS é um *framework* para avaliar acarretamentos entre pares de textos, e o FraCas é um conjunto de dados composto por sentenças com variados tipos de fenômenos linguísticos, como quantificação, anáforas, diferentes leituras de plural, entre outros.

[1] https://tac.nist.gov/

Condoravdi *et al.* (2003), por sua vez, propuseram uma reflexão sobre a importância de considerar a detecção de acarretamentos e contradições como métrica de avaliação para sistemas automáticos. Embora não tenham apresentado um conjunto de dados específico, como no caso do FraCas, os pesquisadores chamaram atenção para o fato de que diferentes tarefas de NLP precisam, em alguma medida, lidar com avaliações de acarretamentos entre textos, palavras ou expressões linguísticas. A questão seria, então, delimitar uma tarefa de modo a torná-la realizável para sistemas automáticos.

Outros trabalhos similares aos citados até aqui podem ser encontrados na literatura em NLP. Com o crescente interesse da comunidade científica sobre a reflexão e importância da validação de acarretamentos textuais, era uma questão de tempo até que um tipo específico de tarefa com esse escopo surgisse. Esse surgimento aconteceu em 2005, com a apresentação do *Pascal RTE Challenge*, quando a NLI ganhou os contornos que hoje conhecemos.

A definição mais difundida na literatura sobre o tema foi apresentada por Dagan *et al.* (2005), organizadores do *Pascal RTE Challenge*, uma primeira tentativa de avaliação padronizada de modelos para NLI. Os organizadores procuraram apresentar uma definição genérica, menos formal e mais orientada pelo escopo de outras tarefas de NLP, como *Information Extraction* (IE) e *Question Answering* (QA). Dagan *et al.* (2005) definem a NLI como:

> [...] uma relação direcional entre pares de textos, denotada por T, o "Texto" acarretante, e H, a "Hipótese" acarretada. Dizemos que T acarreta H se o significado de H pode ser inferido a partir do significado de T, como tipicamente seria interpretado por humanos.[2]

Como podemos perceber, sobretudo pelo segundo período da citação, a verdade de um acarretamento, ou inferência, está baseada no julgamento humano. Definida dessa forma, a NLI aproxima a noção de acarretamento do senso comum e do conhecimento linguístico que humanos possuem. Isso deu aos organizado-

[2] [...] a directional relationship between pairs of text expressions, denoted by T – the entailing "Text", and H – the entailed "Hypothesis". We say that T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people.

res do *Pascal RTE Challenge* a possibilidade de criar conjuntos de dados avaliados conjuntamente por humanos, como já era prática em tarefas de IE e QA.

Em IE e QA é comum que anotadores humanos avaliem relações de acarretamentos entre textos. Dessa forma, é possível obter um padrão de julgamento baseado na concordância de anotadores: o padrão ouro. A especificidade da NLI está no fato de que a noção de acarretamento é capturada de maneira mais genérica e até mesmo independente de outras tarefas em NLP. Idealmente, outras tarefas de NLP que lidem com acarretamentos textuais deveriam beneficiar-se de modelos e ferramentas eficazes desenvolvidas para a NLI.

Além da definição menos formal, outra distinção merece destaque: a noção conceitual de acarretamento, ou inferência, adotada em NLI. Na verdade, os conceitos são equivalentes nessa tarefa e, além disso, distanciam-se de definições mais comuns na literatura linguística e lógica.

Acarretamento, ao menos nas definições mais tradicionais de lógica e semântica, é uma relação na qual a verdade de uma sentença S_2 depende, necessariamente, da verdade de outra sentença S_1. Dito de outra forma, uma sentença S_1 acarreta uma sentença S_2 se, e somente se, sempre que S_1 for verdadeira, S_2 também for. Consideremos o exemplo apresentado em (Jacobson, 2014, p. 32) para ilustrar essa relação:

(1) Mitka matou o pássaro que estava preso na varanda.

(2) O pássaro que estava preso na varanda morreu.

Podemos afirmar que (1) acarreta (2), uma vez que, ao garantirmos a verdade do primeiro exemplo, garantimos, também, a verdade do segundo. Notemos, no entanto, que o acarretamento não seria verdadeiro se considerássemos o inverso, isto é, (2) acarretar (1). O pássaro poderia ter morrido em decorrência de outros eventos ou ações que não a de Mitka.

Essa não é, no entanto, a definição da qual partem os organizadores do *Pascal RTE Challenge*. Ambas, a definição apresentada pelos organizadores e as definições comuns na linguística, estão relacionadas, principalmente quanto à direcionalidade. Porém, em NLI, acarretamentos são definidos de maneira

menos formal, podendo, inclusive, remeter à probabilidade da verdade de uma hipótese dada uma premissa, como distinguido por (Dagan *et al.*, 2013, p. 7):

> Uma definição comum de acarretamento na semântica formal estabelece que um Texto T acarreta outro texto H (hipótese, em nossa terminologia), caso H seja verdadeiro em todas as circunstâncias (mundos possíveis) em que T é verdadeiro. [...] No entanto, a definição de acarretamento textual admite casos em que a verdade da hipótese é muito plausível (muito provavelmente verdadeira), para fins mais práticos do que exatos.[3]

A partir da citação, podemos deduzir que a extensão do conceito de acarretamento em NLI é bem ampla, pois um par de sentenças pode ser anotado como verdadeiro sem que uma relação de consequência lógica exista entre uma premissa e uma hipótese. Esse era, na verdade, o objetivo dos organizadores do desafio. Os conjuntos de dados fornecidos pelo *Pascal RTE Challenge* eram compostos por uma grande variedade de textos, nos quais um acarretamento poderia ser desencadeado por diferentes fenômenos linguísticos.

Outra distinção que merece destaque são os casos em que uma hipótese poderia ser uma tautologia, isto é, a hipótese poderia ser verdadeira em qualquer circunstância. Como sabemos, em uma tautologia, as condições de verdade de uma sentença sempre são satisfeitas, ou seja, não existe a possibilidade de sua falsidade. Na NLI, entretanto, o conteúdo de uma premissa deve ser essencial para a verdade de uma hipótese.

Em verdade, está fora do escopo da NLI julgar o valor de verdade de uma premissa. A premissa deve, portanto, ser tomada como verdadeira. Quanto ao processo de anotação dos pares, as premissas devem ser consideradas como dados seguros e aceitáveis, ao menos no contexto de avaliação.

A anotação dos conjuntos de dados pressupõe, portanto, algum nível de conhecimento linguístico e de conhecimento de mundo compartilhado entre os

[3] A commom definition of entailment in formal semantics specifies that a Text T entails another text H (hypothesis, in our terminology) if H is true in every circumstance (possible world) in which T is true. [...] However, the textual entailment definition allows for cases in which the truth of the hypothesis is highly plausible (most likely true), for most practical purposes, rather than certain.

anotadores. Sendo que apenas com base nesses dois tipos de conhecimento, um par deve ser classificado como verdadeiro ou falso. Portanto, o conhecimento de mundo, sozinho, não poderia justificar uma classificação verdadeira para um acarretamento.

Um exemplo oferecido por (Dagan *et al.*, 2013, p. 7) nos ajuda a ilustrar um cenário característico. Em um possível par com uma premissa do tipo P: "The U.S. citizens elected their new president Obama." e uma hipótese como H: "Obama was born in the U.S.", o senso comum, ou o conhecimento de mundo, por si só não deveria justificar uma classificação "True" para a relação de acarretamento do par. Evidentemente, o senso comum de que é necessário ter nascido nos Estados Unidos para ser presidente estadunidense permite validar a hipótese do exemplo. No entanto, esse tipo de conhecimento não deve ser o único critério para a validação. Ele deve, apenas em combinação com a premissa, justificar o acarretamento. A premissa deve ser, portanto, parte essencial do raciocínio que valida a verdade de uma hipótese, enquanto o conhecimento de mundo não (Dagan *et al.*, 2013).

Normalmente, os conjuntos de dados disponibilizados em NLI são compostos por pares de textos em língua natural. As fontes de extração dos textos variam de acordo com cada corpus e, claro, com o objetivo de cada tarefa. Os exemplos adiante compõem os três primeiros conjuntos de dados do *Pascal RTE Challenge*, o RTE-1, o RTE-2 e o RTE-3, respectivamente:

(3) Id: "822" "Entailment": "TRUE" Task:"CD"
P: Satomi Mitarai died of blood loss.
H: Satomi Mitarai bled to death.

(4) Id:"479" "Entailment": "TRUE" Task: "IE"
P: Google co-founders Sergey Brin and Larry Page apparently have settled on a Boeing 767 as their personal jet.
H: Sergey Brin is a partner of Larry Page.

(5) Id: "115" "Entailment": "FALSE" Task: "IE"
P: Belknap was impeached by a unanimous vote of the House of Representatives for allegedly having received money in return for post tradership

appointments.
H: Belknap received money in return for post tradership appointments.

Os exemplos oferecem a possibilidade de verificar algumas das dificuldades apresentadas pela NLI e, mais particularmente, pelo *Pascal RTE Challenge*. Em uma breve análise, podemos identificar algumas das dificuldades que se impõem ao processamento computacional dos textos, como a presença de entidades nomeadas, por exemplo. Sammons (2015) elenca algumas das competências necessárias para que um humano faça os tipos de inferência exigidos pelos pares apresentados pela NLI: raciocínio temporal e quantitativo, identificação de papéis de entidades em eventos, lidar com ambiguidade, entre outras.

Em (3) faz sentido pensar que, se alguém morreu por perda de sangue, sangrou até a morte, o que justificaria a classificação "TRUE". Reparemos, no entanto, na alternância entre formas verbais e nominais no par, isto é, entre *died/death* e *blood/bled*. Uma classificação automática baseada na co-ocorrência dos *tokens* atribuiria "FALSE" para o par. Seria necessário, portanto, optar por ferramentas automáticas capazes de realizar análises nos níveis sintático, semântico e até mesmo morfológico.

O exemplo (4) possui textos bem díspares em relação ao tamanho. Além disso, as informações apresentadas na premissa divergem bastante das informações apresentadas na hipótese. Pensando na classificação atribuída pelos anotadores, talvez a informação mais relevante para determiná-la seja o fato de que os dois indivíduos mencionados são co-fundadores do Google, o que pressupõe uma parceria profissional entre eles. Do ponto de vista automático, no entanto, a classificação não seria tão simples. Tanto uma análise baseada em alinhamento quanto uma análise baseada na semelhança lexical poderiam induzir um classificador a atribuir uma classificação errada.

Por fim, o exemplo (5) nos permite refletir sobre o tipo de inferência que um humano teria que fazer para atribuir uma classificação "FALSE" ao par. Para além da disparidade entre o tamanho dos textos, as palavras da hipótese estão incluídas na premissa, o que poderia induzir um sistema automático a atribuir uma classificação "TRUE" para o par. Uma única palavra, no entanto, permite inferir que a relação de acarretamento não é verdadeira: *allegedly*.

Os três exemplos apresentados nos oferecem um breve quadro das dificuldades oferecidas pela NLI e, mais especificamente, pelo *Pascal RTE Challenge*. As diferentes possibilidades de associação entre premissas e hipóteses dificultam a classificação automática. Frente ao problema e à complexidade da tarefa, a NLI continua como uma das grandes áreas de investigação em NLP, sendo largamente explorada nas pesquisas atuais.

A solução dos problemas apresentados pela tarefa requer que os modelos candidatos lidem com dificuldades que não são tão simples de automatizar: o conhecimento linguístico e o conhecimento de mundo. O conhecimento linguístico se apresenta como um obstáculo dada a variabilidade de fenômenos linguísticos que podem compor um par, como a variabilidade semântica, sintática, morfológica etc. O conhecimento de mundo, por sua vez, diz respeito ao senso comum e às experiências de vida de humanos.

Desde a criação do *Pascal RTE Challenge* e, consequentemente, da configuração da NLI tal como conhecemos hoje, diferentes tipos de metodologias foram propostas na esperança de solucionar os problemas. Entre as metodologias, estiveram modelos baseados em associações lexicais livres, como o Bag-of-Words (BoW), estratégias probabilísticas, lógicas, diferentes análises linguísticas, metodologias baseadas no aprendizado de máquina e, com frequência, modelos híbridos, isto é, que integram uma ou mais dessas abordagens.

Nesta pesquisa, nosso objetivo foi testar e comparar métodos baseados em técnicas utilizadas, em sua maioria, ao longo dos desafios. Com isso, pretendemos situar este trabalho na discussão mais geral sobre o quanto da solução dos problemas de NLI do *Pascal RTE Challenge* está em cada método testado. Para tanto, procuramos explorar diferentes formas de resolver problemas de NLI oferecidos pelos três primeiros conjuntos de dados do *Pascal RTE Challenge*, o RTE-1, o RTE-2 e o RTE-3. Na próxima subseção, apresentamos algumas das metodologias mais utilizadas na tarefa.

1.1.1 Métodos Propostos Para Solução dos Problemas em NLI

No RTE-1, conjunto de dados do qual retiramos o exemplo (3), temos os trabalhos de Glickman *et al.* (2005), com uma acurácia de 59%, e o de Jijkoun

et al. (2005), com uma acurácia de 56%, como metodologias baseadas em um BoW. Em um modelo de BoW, basicamente, desconsidera-se a estrutura sintática e argumental de uma sentença e, evidentemente, a ordem das palavras. O BoW permite representar uma sentença como uma rede de termos livres e, por meio dessa representação, extrair diferentes informações das palavras, como a frequência absoluta ou relativa, raízes, valores de IDF (frequência inversa de um termo em documentos), entre outras.

Metodologias baseadas em BoW são utilizadas com frequência em diversas áreas de pesquisa em NLP. Quase sempre o BoW é usado em estágios modulares de modelos mais complexos, como os modelos baseados no aprendizado de máquina, por exemplo.

Utilizar um BoW para a tarefa proposta em NLI traz alguns obstáculos, no entanto. Quando se pensa no tipo de problema apresentado, fica evidente que desconsiderar a estrutura argumental das sentenças dos pares significa a perda de informações importantes e, consequentemente, a classificação incorreta de alguns deles. Sentenças que veiculam diferentes informações, mas que possuem as mesmas palavras, acabam tendo uma representação similar, como no exemplo de (MacCartney, 2009, p. 11) "Booth shot Lincoln/Lincoln shot Booth". A língua não é, como bem pontuou Harris (1954), um Bag-of-Words. Por outro lado, o BoW é uma metodologia simples, com pouco custo computacional, o que faz com que esse modelo seja frequentemente utilizado como *baseline* para avaliar metodologias mais complexas.

Propostas de solução baseadas em alinhamento também foram apresentadas ao longo das pesquisas em NLI. Com diferentes métodos para alinhar pares de textos, técnicas de alinhamento se configuraram, de acordo com Dagan *et al.* (2013), como um refinamento de métodos baseados em similaridade, pois possibilitam determinar similaridades locais entre palavras dos textos dos pares.

Métodos de alinhamento são alternativas para lidar com informações sobre a estrutura sintática dos pares, não consideradas pelo BoW, por exemplo. Por meio deles, é possível computar o custo de edição para converter uma premissa em hipótese, selecionar partes específicas de premissas e hipóteses para determinar quais partes dos textos são mais relevantes para a classificação, entre outras possibilidades. No entanto, embora comumente utilizados em tradução automática,

métodos de alinhamento precisam lidar com outro problema representado pela NLI, a saber, a diferença entre o tamanho das premissas e hipóteses. As hipóteses dos pares normalmente são menores do que as premissas, o que pode aumentar o custo do alinhamento. Além disso, ausência de bases de dados para treino de algoritmos de alinhamento também dificulta a eficácia dessas técnicas.

A classificação automática por meio de modelos baseados em aprendizado de máquina supervisionado também foi amplamente utilizada no *Pascal RTE Challenge*. Em metodologias desse tipo, técnicas de aprendizado de máquina são utilizadas para desenvolver modelos estatísticos para classificar se um dado par, por exemplo, é verdadeiro ou falso para a relação de acarretamento.

Um exemplo de modelo para classificação baseado em metodologias de aprendizado de máquina supervisionado é o classificador bayesiano ingênuo, frequentemente utilizado em detecção de *spam*, desambiguação automática e análise de sentimento (Jurafsky & Martin, 2019). Um classificador bayesiano ingênuo permite obter a probabilidade de que determinado dado de entrada pertença a uma determinada classe de um conjunto pré-determinado, no caso da NLI, verdadeiro ou falso para o acarretamento. Trata-se, portanto, de um classificador probabilístico baseado na Regra de Bayes. A equação do classificador bayesiano ingênuo é dada por:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i=1}^{n} P(f_i|c) \tag{1.1}$$

Na fórmula em questão, para obter uma estimativa da classe $\hat{c}$ para um dado conjunto C, é necessário realizar a multiplicação das probabilidades condicionais do i-ésimo atributo f dada uma classe c, ou seja, $P(f_i|c)$. Essa multiplicação de probabilidades é representada pelo símbolo $\prod$. A função argmax possibilitará, então, maximizar o valor do produto entre probabilidade da classe c, identificada por $P(c)$ e o produtório das probabilidades de $P(f_i|c)$.

Com frequência, técnicas de aprendizado de máquina supervisionado também são utilizada em modelos híbridos. Entre as propostas com modelos híbridos, estiveram os trabalhos de Hickl *et al.* (2006) e Tatu *et al.* (2006), que obtiveram 75% e 74% de acurácia no RTE-2, respectivamente. Os primeiros utilizaram o GROUNDHOG, um modelo baseado em alinhamento e aprendizado

de máquina, enquanto os segundos apresentaram o COGEX, um modelo também baseado em aprendizado de máquina e representações lógicas para os pares. Já no RTE-3, com os mesmos modelos, Hickl & Bensley (2007*a*) e Tatu & Moldovan (2007) conseguiram 80% e 72% de acurácia, respectivamente.

Outra proposta de solução apresentada ao longo do *Pascal RTE Challenge* foi representada pelos modelos baseados em representações lógicas. Considerando o problema oferecido pela tarefa, isto é, avaliar acarretamentos, parece natural optar por essa via de solução. Porém, a necessidade de converter para uma forma lógica sentenças com diferentes possibilidades de relações entre si, como subordinação, coordenação, entre outras, pode limitar a performance de modelos baseados essencialmente em metodologias desse tipo.

Para lidar com problemas decorrentes da variabilidade linguística dos pares, modelos baseados em representações lógicas podem aproximar as representações para as palavras utilizando formas lemáticas e bases de conhecimento externo como a WordNet. Representações formais também podem ser componentes de outros modelos, como o já citado COGEX, apresentado Tatu *et al.* (2006) e Tatu & Moldovan (2007).

Apesar dos avanços e das diferentes propostas apresentadas desde seu surgimento, vários obstáculos se impõem à solução dos problemas oferecidos pela NLI. Além do conhecimento linguístico e do conhecimento de mundo, muito mais difícil de automatizar, o tamanho dos conjuntos de dados também pode comprometer o desempenho dos modelos avaliados.

O tamanho dos conjuntos de dados dificulta a eficácia de modelos baseados no aprendizado de máquina, particularmente em redes neurais artificiais. Por outro lado, as possibilidades de associações entre os textos de uma premissa e os textos de uma hipótese, além da variabilidade linguística, representam obstáculos para modelos baseados em associações livres ou na similaridade lexical entre as palavras.

Em anos mais recentes, modelos baseados em redes neurais artificiais com técnicas de *Transfer Learning*[4], ou transferência de aprendizado, têm ganhado cada vez mais espaço em NLP, estabelecendo, inclusive, novos estados da arte.

[4] https://www.tensorflow.org/tutorials/images/transfer_learning

A transferência de aprendizado é um método de aprendizado de máquina no qual um modelo desenvolvido para um tipo tarefa é utilizado em etapas iniciais de outros modelos aplicados em tarefas diferentes. Em outras palavras, é uma técnica utilizada para aplicar modelos pré-treinados em tarefas específicas em outras tarefas de NLP. Esse tipo de método permite, por exemplo, lidar com problemas decorrentes de bases de dados pequenas para etapas de treinamento (Tan *et al.*, 2018).

Entre os modelos baseados nesse tipo de arquitetura, está o BERT (Devlin *et al.*, 2018). O BERT, *Bidirectional Encoder Representation from Transformers*, é um modelo pré-treinado em dados não rotulados, necessitando de uma etapa de *fine tuning* para aplicações específicas em tarefas de NLP. O modelo é baseado em uma arquitetura do tipo *Transformer* (Vaswani *et al.*, 2017) e utiliza o contexto tanto da esquerda para a direita quanto da direita para a esquerda para processar os dados. O pré-treinamento do modelo foi realizado no BookCorpus (Zhu *et al.*, 2015) e na Wikpédia em Inglês.

Atualmente, o BERT é disponibilizado em duas versões[5]: BERT_{base} e BERT_{large}. O BERT_{base} possui 12 camadas do tipo *Transformer*, 768 camadas ocultas e 12 *self-attention heads*, ou mecanismos de atenção. Ao todo, o BERT_{base} conta com 110 milhões de parâmetros. O BERT_{large}, por sua vez, possui 24 camadas do tipo *Transformer*, 1024 camadas ocultas, 16 *self-attention heads* e um total de 340 milhões de parâmetros.

Um dos diferenciais do BERT está no otimização da etapa de *fine tuning* por meio de um mecanismo de mascaramento de tokens, ao qual Devlin *et al.* (2018) chamaram *Masked Language Model* (MLM). O MLM é um mecanismo que permite mascarar aleatoriamente uma porcentagem dos tokens que o modelo recebe como entrada e, assim, realizar a predição dos tokens mascarados. Nos testes realizados por Devlin *et al.* (2018), 15% dos tokens para os quais o modelo gerou vetores foram aleatoriamente mascarados. Além do MLM, o BERT também conta com o *Next Sentence Prediction* (NSP), mecanismo para prever se, dado um par de sentenças, a segunda é subsequente à primeira. O NSP pode ser utilizado, por exemplo, em tarefas como *Question Answering* e a própria NLI.

[5] https://github.com/google-research/bert

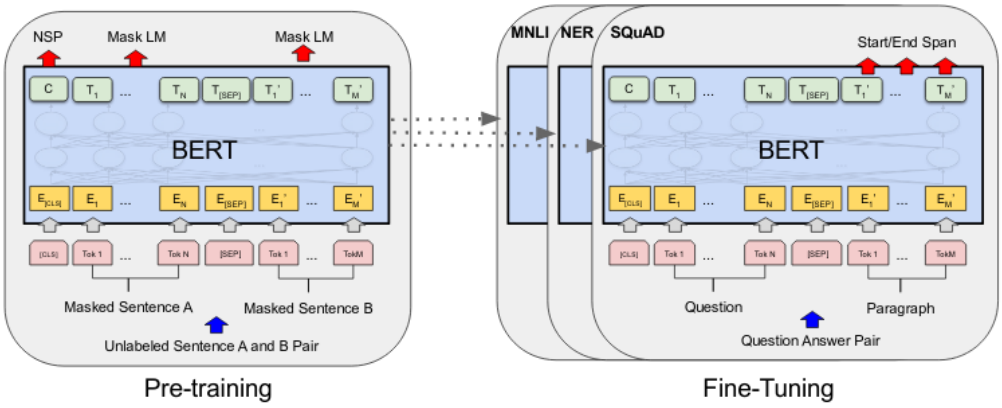


Figura 1.1: Arquitetura do BERT. Fonte: Devlin *et al.* (2018).

A Figura 1.1 oferece uma ideia geral da arquitetura do BERT e também de sua aplicabilidade em diferentes tarefas de NLP. Por meio dela, vemos que o modelo recebe como *input* uma sequência de *tokens* que pode ser composta por uma única sentença ou por mais de uma. O primeiro *token* de uma dada sequência é sempre representado como CLS. O *token* SEP, por sua vez, é utilizado para distinguir um par de sentenças diferentes que podem ou não ser subsequentes. Para os *tokens* fornecidos como *input*, o modelo gera representações vetoriais baseadas em seus contextos de ocorrência. Devlin *et al.* (2018) utilizaram o WordPiece embeddings para gerar vetores durante a etapa de pré-treinamento do BERT. A partir de representações vetoriais geradas para o *input*, o modelo prevê o próximo dado de uma sequência por meio do mecanismo de mascaramento, ou MLM.

Como podemos observar Figura 1.1, o BERT é aplicável a mais de uma tarefa de NLP. Algumas delas são o Reconhecimento de Entidades Nomeadas, do inglês (NER), o Question Answering e a NLI. Em uma tarefa como a NLI, a classificação poderia ser feita por meio do mecanismo de NSP. Outra forma de classificação para os acarretamentos poderia ser realizada por meio do cálculo do grau de similaridade entre as representações vetoriais geradas pelo modelo para um dado par.

Por ser um modelo mais recente, o BERT não foi avaliado no *Pascal RTE Challenge*. Atualmente, no entanto, há um conjunto de dados que engloba

diferentes corpora de tarefas em NLP, entre elas a NLI. Trata-se do GLUE[6], que oferece um conjunto de dados composto pelo RTE-1, RTE-2, RTE-3 e RTE-5. Avaliado nesse conjunto de dados, o BERT obteve uma acurácia de 70%.

Os métodos brevemente descritos nesta subseção nos dão um panorama das propostas de solução para os problemas oferecidos pela NLI. O interesse nessa tarefa decorre da dificuldade em encontrar um consenso sobre a melhor metodologia a usar. Ao longo dos últimos anos, a NLI ganhou novos conjuntos de dados e também novas tarefas compartilhadas na tentativa padronizar uma avaliação para os modelos. Apesar dos avanços e dos resultados obtidos com os modelos mais recentes, como o BERT, essa tarefa ainda permanece aberta em NLP.

1.1.2 Organização dos Capítulos

No Cap. 2, traçamos um breve percurso histórico da NLI, com uma maior ênfase no *Pascal RTE Challenge*. Nesse capítulo, apresentamos uma breve revisão de literatura abordando os oito desafios apresentados ao longo do *Pascal RTE Challenge*, alguns novos conjuntos de dados para NLI, algumas discussões sobre o escopo e os conjuntos de dados da tarefa e, por fim, algumas das principais bases de conhecimento externo utilizadas. O objetivo deste capítulo é contextualizar o leitor sobre o desenvolvimento da tarefa desde seu surgimento até as pesquisas mais atuais. O Cap. 3 apresenta os objetivos centrais do trabalho. Nele, procuramos delinear as questões que guiaram esta pesquisa e nossos testes. Nossa metodologia é apresentada no Cap. 4. Nele apresentamos os conjuntos de dados selecionados, descrevemos a etapa de pré-processamento desses conjuntos e os pseudocódigos de cada teste realizado. O objetivo do capítulo é delinear as etapas práticas da pesquisa de modo a torná-las reprodutíveis. O Cap. 5 apresenta nossos resultados. Nesse capítulo, discutimos a avaliação dos testes realizados e os comparamos. Por fim, no Cap. 6, apresentamos a conclusão do trabalho. Nosso intuito, nesse capítulo, é apresentar uma reflexão sobre a proposta do trabalho, seus pontos fortes e fracos.

[6] https://gluebenchmark.com/

2

Revisão de Literatura

Neste capítulo, traçamos um breve histórico da NLI dando maior atenção ao *Pascal RTE Challenge* em uma tentativa de melhor contextualizar o leitor sobre o desafio que marca o início da tarefa. Ao longo do capítulo, apresentaremos um panorama geral sobre algumas das metodologias mais utilizadas, além de algumas discussões e críticas à tarefa. Apresentaremos, também, outros conjuntos de dados e alguns dos recursos externos mais utilizados na NLI.

2.1 Breve Histórico do Pascal RTE Challenge

Como apresentamos na Seção 1 do capítulo anterior, o *Pascal RTE Challenge* surgiu em 2005 com Dagan *et al.* (2005) como uma primeira tentativa de avaliação padronizada de modelos para NLI. Na ocasião, os criadores apresentaram um novo conjunto de dados que viria a ser o primeiro de uma série de oito para avaliação das mais variadas metodologias em uma tarefa anualmente compartilhada. Inicialmente, a série de desafios foi patrocinada pela *European Pascal Network of Excellence* e, a partir de sua quarta edição, ligada à *Text Analysis Conference*, organizada pelo *National Institute of Standard and Methods* do governo dos Estados Unidos (Dagan *et al.*, 2013).

Em suas primeiras três edições, os pares recebiam uma classificação binária, isto é, verdadeiro-falso para a relação de acarretamento. A partir do RTE-4, a classificação de "contradição" foi incluída. A definição de contradição, baseada em Voorhees (2008, p. 64), estabelecia que um par deveria ser avaliado como contraditório para o acarretamento, caso uma humano pudesse concluir que a hipótese seria muito provavelmente falsa dada uma premissa, que sempre é as-

sumida como verdadeira na tarefa, ou seja, ambas não poderiam ser verdadeiras ao mesmo tempo.

Os dados foram retirados, em sua maioria, de fontes textuais do gênero "notícias". De acordo com Bentivogli *et al.* (2017), com exceção do RTE-1, que teve dois anotadores, todos os outros corpora foram anotados por ao menos três anotadores. A principal métrica para a avaliação dos modelos foi a acurácia. Posteriormente, após o RTE-6, os modelos passaram a ser avaliados pelo cômputo da medida-F.

De uma maneira geral, os conjuntos de dados do *Pascal RTE Challenge* possuem uma grande variedade de fenômenos linguísticos que dificultam a classificação automática. Bentivogli *et al.* (2010*a*) apresentaram a descrição de uma metodologia para criar um conjunto de dados específico para classificação de acarretamentos a partir de 90 exemplos do RTE-5. O objetivo dos autores era criar um conjunto de dados com fenômenos linguísticos isolados e necessários para a classificação de um acarretamento. Após a tentativa, os autores caracterizaram cinco macrocategorias de fenômenos linguísticos presentes no RTE-5 e nos RTEs anteriores: (i) lexical, como sinonímia, hiperonímia etc.; (ii) léxico-sintático, como nominalizações; (iii) sintáticos, como alternância de estrutura argumental; (iv) discursiva, como anáforas e (v) raciocínio temporal e espacial.

De fato, os proponentes do desafio procuraram criar conjuntos de dados que fossem repletos dos mais variados fenômenos linguísticos, o que tornou os RTEs difíceis de classificar. As várias possibilidades de associação entre uma premissa e uma hipótese dificultavam o bom desempenho dos modelos avaliados.

A criação dos conjuntos de dados foi, basicamente, dividida em duas etapas: (i) coleta e criação dos pares; e (ii) anotação. Em um primeiro momento, os pares foram gerados e anotados para a relação de acarretamento. Posteriormente, foram avaliados por anotadores e, por fim, os criadores de cada desafio filtraram pares nos quais não houve concordância entre os anotadores.

Os pares dos corpora possuem itens que dificultam a classificação automática, como as entidades nomeadas (itens que denotam pessoas, instituições, dias da semana), expressões temporais, entre outros. Ao longo dos desafios, o tamanho dos corpora variou de 800 a 1.000 pares.

Até o RTE-5, os pares eram identificados com o tipo de tarefa para a qual a classificação do acarretamento seria aplicável. Os pares do RTE-1 foram identificados com sete tipos de tarefa: *Question Answering* (QA), *Machine Translation* (MT), *Information Extraction* (IE), *Information Retrieval* (IR), *Comparable Documents* (CD), *Reading Comprehension* (RC) e *Paraphrase Acquisition* (PP). A partir do RTE-2, porém, os tipos de tarefas diminuíram, sendo quatro, no total: IE, IR, QA e *Multi-document Summarization* (SUM), similar à CD.

Para a tarefa de CD, os organizadores utilizaram conjuntos de artigos de jornais que relatassem uma mesma notícia. Os pares de sentenças foram, então, extraídos e classificados em verdadeiro ou falso para a relação de acarretamento. De acordo com Dagan *et al.* (2005), por serem extraídos de notícias sobre um mesmo assunto, pares identificados com essa tarefa tendem a ser lexicalmente semelhantes.

Os pares para SUM foram criados de modo relativamente similar aos de CD. Assim como em CD, os pares para SUM foram retirados de documentos similares. No entanto, foram extraídos com ferramentas de sumarização automática. Cabia aos anotadores selecionar os pares a partir dos documentos retornados pelas ferramentas automáticas.

Na criação dos pares de RC, os anotadores formularam hipóteses relativas a textos, de modo a simular uma avaliação de compreensão de leitura de estudantes. Embora Dagan *et al.* (2005) afirmem que a criação desses pares deveria corresponder a uma situação tipica de avaliação estudantil, não especificam qual nível estudantil deveria servir de base para a criação, o que possivelmente poderia influenciar na complexidade dos textos.

Para criar os pares de QA, os anotadores utilizaram ferramentas automáticas próprias para sistemas de pergunta-resposta. As premissas foram criadas a partir de textos fornecidos como respostas corretas para determinadas perguntas. As hipóteses, por sua vez, foram criadas a partir da conversão das perguntas em sentenças declarativas, que deveriam ser concatenadas a trechos das respostas fornecidas pelos sistemas automáticos.

Dagan *et al.* (2010), oferecem um exemplo do procedimento realizado pelos anotadores na criação desses pares. Em uma questão como “How many inhabitants does Slovenia have?” e uma resposta automática como “In other words,

with its 2 million inhabitants, Slovenia has only 5.5 thousand professional soldiers", os anotadores deveriam utilizar a resposta automática como premissa. Para a hipótese, deveriam converter a pergunta em uma sentença declarativa e concatená-la ao trecho com a resposta para a pergunta, no caso "2 million", o que resultaria em uma hipótese com a seguinte sentença "Slovenia has 2 million inhabitants".

Os pares de MT foram criados a partir de duas traduções de um mesmo trecho, uma automática e uma humana, considerada o padrão ouro. As traduções automáticas foram gramaticalmente ajustadas quando necessário. Tanto premissas quanto hipóteses poderiam ser traduções automáticas ou humanas. Segundo os criadores do desafio, traduções automáticas consideradas corretas deveriam caracterizar a classificação verdadeira para o acarretamento.

Na criação dos pares de IE, os anotadores deveriam considerar um cenário típico de extração de informação e utilizar trechos de textos de notícias em que fosse possível encontrar relações com o cenário. Os trechos serviram para compor as premissas. As hipóteses, por sua vez, foram manualmente construídas com base em cada cenário utilizado para buscar trechos de texto.

Um exemplo de procedimento para a criação de pares desse tipo é oferecido por Dagan *et al.* (2005): dada a tarefa de extrair trechos de textos com base no cenário "killing of civilians", os anotadores poderiam selecionar como premissa um trecho do tipo "Guerrillas killed a peasant in the city of Flores". A hipótese seria, então, manualmente construída como "Guerrillas killed a civilian".

Para os pares de IR, os anotadores criaram hipóteses que serviram como expressões de busca para ferramentas de recuperação de informação. As expressões de busca, obviamente, eram maiores do que palavras-chave utilizadas em cenários de recuperação de informação e foram selecionadas com base em notícias de destaque em jornais. As premissas, por sua vez, foram selecionadas com base nos trechos retornados pelas ferramentas de busca.

Por fim, na criação dos pares de PP, os anotadores selecionaram trechos de notícias e os forneceram para ferramentas automáticas que produziram um conjunto de paráfrases. As paráfrases geradas foram utilizadas como hipóteses que poderiam, ou não, ser acarretadas pelas premissas.

2.2 Os Oito Desafios do Pascal RTE Challenge

Como apresentamos na seção anterior, o *Pascal RTE Challenge* teve oito edições anuais. Em cada um de seus anos, um conjunto de dados foi disponibilizado para que diferentes metodologias fossem avaliadas. O desenvolvimento do desafio chamou atenção da comunidade científica em NLP, que passou a dedicar esforços para apresentar soluções aos problemas. À medida que a NLI ganhava protagonismo nas pesquisas em NLP, no entanto, o RTE também ia se modificando.

Aos poucos, sobretudo a partir do RTE-6, o desafio foi ganhando novas configurações. A avaliação de acarretamentos foi direcionada para um único tipo de tarefa, como sumarização automática no RTE-6 e no RTE-7 e para sistemas de diálogo, já no RTE-8. Do sexto desafio para o oitavo, embora a tarefa central ainda fosse avaliar acarretamentos, o *Pascal RTE Challenge* já não era, essencialmente, similar ao que fora em seus primeiros anos.

O RTE-1, primeiro conjunto de dados do desafio, é composto por um conjunto de teste com 800 pares e um conjunto de validação com 560 pares. Apresentado por Dagan *et al.* (2005), surgiu como um corpus que estimularia o desenvolvimento de ferramentas úteis a diversas áreas de NLP que lidavam com avaliação de acarretamentos.

O corpus possui uma classificação binária, verdadeiro-falso, e é balanceado em relação à classificação de acarretamento, ou seja, 50% dos pares são classificados como verdadeiros e 50% como falsos. Os tipos de tarefa que identificam os pares são sete: QA, IE, IR, PP, RC, MT e CD. Assim como a classificação dos acarretamentos, as tarefas foram igualmente distribuídas entre pares verdadeiros e falsos.

O melhor resultado no primeiro ano do desafio foi obtido por Delmonte *et al.* (2005), com uma acurácia de 60%. Os pesquisadores apresentaram o VENSES (Venice Semantic Evaluation System), um modelo que possuía um sistema de recompensa e penalidade para a similaridade lexical. Palavras lexicalmente similares recebiam, evidentemente, uma taxa de recompensa, palavras sem similaridade, recebiam um valor de penalidade. A classificação era feita por meio

da comparação entre os valores obtidos com o cômputo das penalidades e um limiar estabelecido manualmente.

Com o objetivo de dar continuidade ao *Pascal RTE Challenge*, Bar-Haim *et al.* (2006) apresentaram o RTE-2, segundo corpus de NLI para o desafio. Sua criação foi guiada pela ambição de fornecer um corpus mais realista do que o RTE-1. Assim como o RTE-1, o RTE-2 possui 800 pares para teste. Seu conjunto de validação, porém, foi um pouco maior, com 800 pares.

As novidades, no entanto, não foram muitas, o que fez do RTE-2 um corpus muito parecido com o RTE-1. As premissas continuaram maiores do que as hipóteses, com aproximadamente duas sentenças, e a classificação verdadeiro-falso também foi balanceada em 50%. Os tipos de tarefas, porém, foram diminuídos para quatro: IE, IR, QA e SUM.

Os melhores resultados foram obtidos por Hickl *et al.* (2006), com o GROUNDHOG System, modelo que obteve 75% de acurácia. O modelo apresentado pelos pesquisadores contou com etapas de alinhamento e detecção de paráfrases, além de módulos de processamento linguístico. Para contornar possíveis problemas em decorrência da ausência de base de dados para treinamento do modelo, os pesquisadores criaram conjuntos de dados à parte para treinar os algoritmos de alinhamento e detecção de paráfase.

O RTE-3 foi um pouco mais inovador em relação aos anteriores. De acordo com Giampiccolo *et al.* (2007), o objetivo das inovações foi estimular a participação tanto de novas quanto de antigas equipes. A complexidade da tarefa e a necessidade de utilização de diferentes ferramentas de NLP demonstravam a necessidade de estímulo ao debate e troca de conhecimento entre os participantes.

A primeira inovação foi em relação às premissas, que em alguns pares chegaram ao tamanho de um parágrafo[1]. Com isso, Giampiccolo *et al.* (2007) pretendiam fornecer pares que caracterizassem cenários mais realistas e necessitassem até mesmo de análises no nível discursivo. A segunda inovação foi representada pela criação de um ambiente virtual de compartilhamento. Com

[1] A noção de parágrafo adotada por Giampiccolo *et al.* (2007) é bem ampla. Normalmente, os pares do RTE-1 e do RTE-2 possuem premissas com um ou dois períodos. Ter o tamanho de um parágrafo pode significar, portanto, ter mais do que dois períodos.

isso, esperava-se estimular os participantes a compartilharem informações sobre diferentes recursos e ferramentas automáticas utilizadas ao longo do desafio.

Assim como nos desafios anteriores, o conjunto de teste e o conjunto de validação tiveram 800 pares cada. Aproximadamente 17% dos pares possuíam premissas com o tamanho de um parágrafo. Os tipos de tarefa também foram quatro e os procedimentos de criação dos pares foram os mesmos do RTE-2. A classificação dos acarretamentos, no entanto, não ficou balanceada, embora esse fosse o objetivo de Giampiccolo *et al.* (2007). Tanto no conjunto de validação quanto no conjunto de teste, o número de pares falsos foi maior, aproximadamente 51,50% e 51,25% respectivamente. Os melhores resultados foram conseguidos por Hickl & Bensley (2007*b*), com um modelo similar ao do ano anterior, que alcançou uma acurácia de 80%.

O RTE-3 contou ainda com um projeto piloto chamado *Extending the Evaluation of Inferences from Texts*. O projeto pretendia explorar duas outras tarefas relacionada à avaliação de acarretamento: classificar como 'UNKNOWN' pares em que o acarretamento fosse incerto, distinguindo-os dos pares contraditórios, e fornecer uma justificativa para a classificação dos modelos.

O que era um projeto piloto no RTE-3 foi implementado no RTE-4. O conjunto de dados passou a ter três possíveis classificações: verdadeiro, falso e indefinido. Outra inovação foi o trabalho em conjunto do *National Institute of Standards and Techonology* e do CELCT, apresentando o RTE como um desafio vinculado à *Text Analysis Conference*. Segundo Giampiccolo *et al.* (2008), a junção dos esforços das duas agências proporcionaria maior qualidade ao corpus e também mais recursos para os participantes.

O tamanho do corpus aumentou, passando a ser de 1.000 pares para o conjunto de teste. Porém, no RTE-4, os proponentes não disponibilizaram um conjunto de validação. De acordo com Giampiccolo *et al.* (2008), os pares do RTE-4 eram similares aos dos conjuntos de dados dos anos anteriores e, portanto, os participantes poderiam utilizar os três primeiros corpora para treinar seus respectivos modelos.

Os procedimentos de coleta dos pares, atribuição de tipos de tarefa e anotação foram os mesmos dos anos anteriores. Dessa vez, porém, os tipos de tarefa IE e IR foram distribuídos para 300 pares cada. Em relação à classificação dos acarreta-

mentos, 50% dos pares foram classificados como verdadeiros, 35% como falsos e 15% como indefinidos. O corpus final ainda foi revisado por falantes nativos da língua inglesa para retirar erros de grafias e erros gramaticais produzidos pelas ferramentas automáticas. Os participantes poderiam submeter resultados para a classificação binária ou ternária. Um dos melhores resultados, 65% de acurácia na classificação binária, foi obtido por Mohammad *et al.* (2008), com um modelo baseado em sumarização automática.

O quinto desafio deu continuidade a algumas das mudanças implementadas no RTE-4. Segundo Bentivogli *et al.* (2009), o objetivo da continuidade foi facilitar a comparação da evolução dos modelos testados nos anos anteriores. O tamanho das premissas, que no ano anterior foi maior do que a média dos primeiros RTEs, aumentou ainda mais, chegando a uma média de 100 palavras. Ao contrário do RTE-4, que teve falantes nativos corrigindo erros gramaticais e de digitação, no RTE-5 os textos retornados por ferramentas automáticas não foram corrigidos. Lidar com erros gramaticais e de digitação foi mais um desafio apresentado aos modelos avaliados.

Cada tipo de tarefa, os mesmos quatro do ano anterior, foi atribuído a 400 pares, ou seja, o corpus teve 1.200 pares divididos em conjunto de validação e conjunto de teste. O RTE-5 é, portanto, um corpus menor do que o RTE-4. A proporção de pares classificados como verdeiro foi de 50%, como indeterminado foi de 35% e como falso foi de 15%. Os melhores resultados foram obtidos por Wang *et al.* (2009), com uma acurácia de 68%.

No sexto ano do desafio, Bentivogli *et al.* (2010*b*) apresentaram o RTE-6. Nesse desafio, no entanto, houve uma mudança no direcionamento da tarefa. Dessa vez, ao invés de avaliar o acarretamento entre um par isolado de premissa e hipótese, o desafio consistiu na avaliação dentro de um conjunto de textos. Para tanto, o RTE-6 se configurou como uma tarefa de sumarização automática, propriamente. De acordo com Bentivogli *et al.* (2010*b*), o objetivo era estimar a contribuição que o RTE poderia oferecer para esse tipo de tarefa. O problema consistiu em, dada uma hipótese qualquer e um conjunto de textos, avaliar quais premissas desse conjunto acarretariam a hipótese fornecida.

O RTE-6 foi baseado no TAC 2009 Update Summarization Task[2], um corpus para sumarização automática que é composto por um determinado número de tópicos, cada um contendo dois conjuntos de documentos: o conjunto A, os dez primeiros textos em ordem cronológica de publicação, e o conjunto B, os dez últimos textos. O RTE-6 foi composto por vinte tópicos, sendo que dez foram usados para o conjunto de teste e dez para o conjunto de validação. Uma das particularidades do RTE-6 é a presença de hipóteses que podem ser acarretadas por mais de uma premissa.

O PKTUM, modelo apresentado por Jia *et al.* (2010), obteve o melhor resultado no RTE-6. Com 48% de medida F, o modelo nos permite estimar a dificuldade apresentada pelo desafio. O PKTUM era um sistema capaz de gerar árvores de dependências para as palavras de um determinado texto. A intenção dos autores era verificar nós em partes das hipóteses e compará-los a nós nas premissas. Além da abordagem baseada na análise de dependências, os autores utilizaram também recursos lexicais, como a WordNet e o VerbOcean, etapas de análise morfológica, reconhecimento de entidades nomeadas e normalização de expressões temporais.

O RTE-7 Bentivogli *et al.* (2011) deu continuidade ao que foi implementado no ano anterior, ou seja, oferecer um corpus que aproximasse a NLI da sumarização automática. Desta vez, porém, o corpus foi construído a partir do TAC 2008 e 2009 Update Summarization Task, sendo composto por 20 conjuntos de textos, 10 para o conjunto de teste e 10 para o conjunto de validação.

No RTE-7, os melhores resultados foram obtidos por Tsuchida & Ishikawa (2011), com uma medida-F de 48%. O modelo implementado pelos pesquisadores foi baseado principalmente na semelhança lexical. Para a avaliação dos acarretamentos, computaram o peso informacional das palavras da hipótese que ocorriam na premissa. Além disso, o modelo contou também com o CatVar[3] e com a WordNet como recursos para identificar co-ocorrências e relações lexicais entre premissas e hipóteses.

[2] https://tac.nist.gov//2009/Summarization/update.summ.09.guidelines.html

[3] https://clipdemos.umiacs.umd.edu/catvar/

O RTE-8 Dzikovska *et al.* (2013), último corpus lançado para o desafio, foi proposto como uma tarefa conjunta ao SemEval-2013[4]. O objetivo, segundo Dzikovska *et al.* (2013), era verificar como metodologias utilizadas para avaliação de acarretamentos textuais poderiam contribuir com os avanços da *Student Response Analysis* (SRA). A SRA consiste na tarefa de classificar respostas de estudantes para sistemas automáticos de diálogos.

Levy *et al.* (2013) apresentaram uma metodologia híbrida no último dos desafios. O modelo dos autores utilizava métodos de BoW, além de informações semânticas e sintáticas para alimentar um classificador de árvore de decisão. A medida-F do modelo ficou em 63%. Outro resultado de destaque no RTE-8 foi o obtido por Ott *et al.* (2013). Esses pesquisadores apresentaram o Comet, um classificador composto por três níveis de processamento de textos: duas etapas de alinhamento e uma com regras de um BoW. A medida-F do modelo ficou em 71%.

2.3 Outros Conjuntos de Dados para NLI

À medida que a NLI foi ganhando espaço e se consolidando como um tópico de pesquisa em NLP, novos corpora foram criados, cada um com características próprias e, algumas vezes, com finalidades diferentes, como possibilitar a utilização de metodologias que exigem grande quantidade de dados para treino. Nesta seção alguns corpora são brevemente apresentados. O objetivo da descrição não é expor aqueles que são considerados os melhores ou principais corpora em NLI, mas apenas demonstrar sua existência e importância para a continuidade das pesquisas e debates sobre o tema. Cada um dos corpora apresentados contribuiu e continua contribuindo para os avanços, questionamentos e melhor delineamento do escopo da tarefa.

[4] https://www.cs.york.ac.uk/semeval-2013/task7/index.html

2.3.1 O FraCas Test Suite

O FraCas Teste Suite Cooper *et al.* (1996) é um conjunto de dados em inglês apresentado pelo FraCas Consortium, grupo que também apresentou o CLEARS, um *framework* para avaliar acarretamentos. No total, o corpus possui 346 problemas para NLI compostos por uma ou mais sentenças seguidas por uma pergunta e uma resposta. Em relação à classificação, as respostas das perguntas podem ser classificadas como verdadeira, contraditória ou neutra.

Uma das principais diferenças entre o FraCas e os demais conjuntos de dados para NLI é a forma como o problema é apresentado. Na NLI, como vimos anteriormente, normalmente premissas e hipóteses são sentenças declarativas, enquando no FraCas o problema é apresentado em forma de questões sobre as premissas. As sentenças do FraCas são relativamente similares. No entanto, diferentes fenômenos linguísticos são apresentados, como anáforas, quantificação, referência temporal, elipses, entre outros.

MacCartney (2009) avaliou seu modelo, o Natlog System, nesse conjunto de dados. Para tanto, converteu as questões em hipóteses declarativas com uma ferramenta de análise sintática e, posteriormente, realizou uma análise manual para corrigir erros. O modelo do autor obteve uma acurácia de 70% no corpus.

2.3.2 O SICK

O SICK, *Sentence Involving Compositional Knowledge*, é um corpus em inglês disponibilizado por Marelli *et al.* (2014) para avaliar modelos de NLI baseados em semântica composicional, especificamente modelos de semântica distribucional. Segundo os autores, corpora como os do *Pascal RTE Challenge* exigem metodologias capazes de lidar com fenômenos linguísticos fora do escopo teórico da semântica composicional, como expressões idiomáticas, entidades nomeadas, dígitos, entre outros.

O corpus possui 10.000 pares de sentenças (premissas e hipóteses) anotados quanto à relação de acarretamento e ao grau de semelhança entre o conteúdo semântico dos textos. Para a classificação de acarretamento os pares são anotados em relação a três tipos de classificação: acarretamento, contradição e neutralidade. Para o grau de semelhança semântica, por sua vez, os pares receberam

uma avaliação em uma escala de 1 a 5. A anotação foi feita através do Amazon Mechanical Turk [5] e do CrowdFlower, atualmente Figure-Eight[6].

Os pares do SICK foram criados a partir de imagens de atividades cotidianas extraídas de dois outros corpora: o 8K ImageFlickr e o SemEval 2012 STS MSR-Video Description. Em uma etapa de normalização, os criadores do SICK retiraram expressões linguísticas que exigissem análises fora do escopo da semântica composicional, como entidades nomeadas, expressões idiomáticas e expressões temporais. Há também uma versão do SICK para o português brasileiro apresentada por Real *et al.* (2018). Nesse trabalho, os pesquisadores traduziram os pares do SICK em inglês para o português brasileiro mantendo a classificação e os mesmos fenômenos linguísticos corpus original.

2.3.3 O SNLI

O SNLI, *Stanford Natural Language Inference*, é um corpus em inglês criado para suprir a necessidade de uma base de dados com tamanho suficiente para treinar algumas metodologias utilizadas em NLP, como redes neurais artificiais (Bowman *et al.*, 2015). Ao todo, o corpus possui 570.152 pares que estão anotados para um dos três tipos de classificação: acarretamento, contradição e neutralidade.

Os anotadores, Mechanical Turkers, deveriam criar hipóteses que fossem neutras, contraditórias ou acarretadas por sentenças extraídas de fotografias do corpus Flickr30K[7]. É importante ressaltar que os anotadores não tiveram acesso às fotografias. Para medir a qualidade do corpus, 10% dos pares foram separados e fornecidos para quatro novos anotadores que deveriam, então, avaliar a classificação. Os pares foram validados caso pelo menos três anotadores concordassem sobre a classificação, o que incluía também os criadores das hipóteses.

Pela quantidade de pares, o SNLI oferece a possibilidade de testar metodologias que necessitem de grande quantidade de dados para treino. Segundo Bowman *et al.* (2015), metodologias desse tipo, como as baseadas em redes neurais

[5] https://www.mturk.com/

[6] https://www.figure-eight.com/

[7] http://bryanplummer.com/Flickr30kEntities/

artificiais, não haviam alcançado resultados expressivos em outros corpora de NLI devido à limitações no tamanho das bases de dados. Diferente dos RTEs, o SNLI não fez parte de nenhuma avaliação conjunta. Desde a sua criação, no entanto, o corpus permanece sendo amplamente utilizado pela comunidade científica.

2.3.4 O MultiNLI

O MultiNLI, *Multi-Genre NLI Corpus*, é um corpus em inglês disponibilizado por Williams *et al.* (2017). Assim como o SNLI, também foi criado para fornecer uma base de dados com tamanho suficiente para modelos baseados em redes neurais artificiais. A proposta dos pesquisadores, porém, era oferecer um conjunto de dados com textos mais diversificados. Segundo os autores, embora o SNLI ofereça uma grande base de dados para treino de algumas metodologias, não seria ideal para avaliá-las por duas razões: (i) as premissas do SNLI são extraídas de um único gênero textual, são limitadas à descrição concreta das cenas e as hipóteses são curtas e simples; (ii) como consequência de (i), o corpus não oferece um banco de dados efetivo capaz de generalizar a performance dos modelos de NLI e avaliar a compreensão computacional da linguagem humana.

A criação e anotação dos pares foi feita de maneira similar ao SNLI. As premissas foram extraídas de dez fontes diferentes, sendo que os anotadores, Mechanical Turkers, deveriam criar as hipóteses. O MultiNLI também possui três tipos possíveis de classificação: acarretamento, contradição e neutralidade. A avaliação das classificações, como no SNLI, foi feita a partir do julgamento de mais quatro anotadores. Caso pelo menos três anotadores concordassem na classificação, o par era validado para compor o corpus.

Ao todo, o MultiNLI possui 533 mil pares, ou seja, é um corpus um pouco menor do que o SNLI. Apesar disso, os criadores afirmam que seu diferencial do está na diversidade dos pares, que representam variedades da modalidade escrita e oral do inglês americano. Em 2017, o MultiNLI foi utilizado no RepEval 2017, um evento para avaliação conjunta de modelos baseados em redes neurais artificiais (Nangia *et al.*, 2017).

2.3.5 O SciTail

O SciTail, apresentado por Khot *et al.* (2018), é um conjunto de dados em inglês que foi desenvolvido a partir de perguntas de múltipla escolha sobre ciências em textos escolares. As premissas foram extraídas de textos da Web. As hipóteses, por sua vez, foram criadas combinando as perguntas e as respostas corretas e transformando-as em sentenças assertivas.

O conjunto de dados, composto por 27.000 pares, foi criado para aproximar a NLI de tarefas como a QA. De acordo com Khot *et al.* (2018), o corpus foi desenvolvido com objetivo de oferecer desafios que possibilitassem estimar o que um bom sistema de QA precisaria para realizar uma inferência.

Em relação à anotação, o SciTail possui classificações de acarretamento e neutralidade. Os pares foram anotados por cinco participantes, sendo mantidos apenas pares com ao menos 80% de concordância para a relação de acarretamento ou neutralidade. Na época de sua criação, o pesquisadores caracterizaram o corpus como desafiador para modelos avaliados em outros corpora, como o SNLI.

2.3.6 O ASSIN

Embora a NLI tenha ganhado cada vez mais importância em NLP e tenha se expandido como campo de pesquisa, a maioria dos conjuntos de dados disponibilizados para a tarefa são em inglês. Nos últimos anos, no entanto, novas propostas foram apresentadas em outras línguas, contribuindo ainda mais para o crescimento da tarefa. É o caso do ASSIN, conjunto de dados apresentado por Fonseca *et al.* (2016) para o português europeu e também para o português brasileiro.

Na criação dos pares, os pesquisadores utilizaram *Latent Dirichlet Allocations* (LDA). Dois modelos diferentes de LDA foram utilizados, um para o português brasileiro, treinado no site de notícias G1[8], e outro para o português europeu, treinado no jornal Público[9]. Após a etapa de treinamento, os auto-

[8] https://g1.globo.com/

[9] https://www.publico.pt/

res utilizaram os modelos para extrair pares similares do site *Google News* nas versões específicas para o Brasil e Portugal.

Quatro pessoas anotaram cada par, atribuindo um valor de 1 a 5 para a similaridade entre os textos e uma das possíveis classificações: (i) a primeira sentença implica a segunda; (ii) a segunda implica a primeira; (iii) paráfrase, ou nenhuma relação. Os pares também foram manualmente revisados para eliminar sentenças agramaticais e erros gramaticais.

No ano de sua apresentação, seis equipes participaram de uma avaliação conjunta, sendo três brasileiras e três portuguesas. Todas as equipes participaram da tarefa de similaridade, enquanto apenas quatro participaram da avaliação de inferência textual.

Fialho *et al.* (2016) obtiveram os melhores resultados para os dois tipos de tarefas, similaridade e inferência, nas duas línguas. Com um modelo baseado em diferentes métricas de similaridade, como a distância de edição e o BLEU, conseguiram uma medida-F de 70% na tarefa de inferência e um valor de 0,73 para a correlação de Pearson na tarefa de similaridade semântica.

No português brasileiro, Barbosa *et al.* (2017) conseguiram os melhores resultados na tarefa de inferência. Os autores utilizaram Word2Vec para gerar vetores de palavras das sentenças e compará-las. O modelo obteve uma medida-F de 82%.

Fonseca (2018) avaliou dois modelos no corpus, o TEDIN e o Infernal. O primeiro modelo foi uma combinação de uma arquitetura de redes neurais com a distância de edição de árvores sintáticas. O segundo foi baseado em aprendizado de máquina e engenharia de atributos. O TEDIN atingiu aproximadamente 50% de medida-F, enquanto o Infernal obteve 86% para a mesma medida. Em anos mais recentes uma nova tarefa compartilhada foi realizada com o corpus[10], apresentada por Real *et al.* (2020).

[10] https://sites.google.com/view/assin2/

2.4 Metodologias

Ao longo de seus oito desafios, o *Pascal RTE Challenge* foi um espaço de desenvolvimento e testes das mais variadas metodologias. Os resultados alcançados, os progressos e as dificuldades enfrentadas foram aos poucos incorporados a cada nova edição da tarefa. Nesta seção, apresentamos um panorama geral das metodologias avaliadas no desafio e, mais amplamente, na NLI.

Entre as abordagens mais bem sucedidas no primeiro ano do desafio estiveram aquelas baseadas na similaridade entre os textos dos pares. Na literatura sobre o tema, esse tipo de metodologia é com frequência chamado de *Shallow Approach*, ou Abordagem Superficial. As heurísticas utilizadas para verificar a similaridade entre premissas e hipóteses variaram bastante entre maior ou menor sofisticação ao longo dos desafios.

No RTE-1, temos os supracitados trabalhos de Glickman *et al.* (2005), com 59% de acurácia, e Jijkoun *et al.* (2005), com 56% de acurácia, como representantes desse tipo de metodologia. Os primeiros apresentaram uma abordagem probabilística e os segundos utilizaram recursos como a WordNet (Miller, 1995)[11] e o cálculo de similaridade baseado em (Lin *et al.*, 1998).

Na verdade, as próprias características dos conjuntos de dados motivaram o uso de técnicas desse tipo. Os RTEs, até seus últimos anos, foram conjuntos de dados considerados pequenos para modelos baseados no aprendizado de máquina e em redes neurais artificiais. A similaridade lexical, por sua vez, permaneceu como um traço característico dos pares até pelo menos o RTE-4. Assim, sendo um parâmetro central ou não na arquitetura dos modelos, técnicas de avaliação de similaridade lexical sempre estiveram presentes.

Outra técnica bastante utilizada no RTE e mais amplamente na NLI foi o alinhamento. Embora esse tipo de técnica seja com frequência baseada na coincidência entre pares de sentenças, possui refinamentos que minimizam problemas decorrentes da variabilidade sintática dos pares.

Como metodologias que utilizaram técnicas de alinhamento, temos Kouylekov & Magnini (2005), pesquisadores que utilizaram uma abordagem de *Tree*

[11] https://wordnet.princeton.edu/

Edit Distance (TED), técnica que permite estimar o custo de edição entre duas sentenças com base em operações como substituição, inserção ou remoção de *tokens*. No RTE-1, os pesquisadores conseguiram 55% de acurácia.

Outras metodologias baseadas em alinhamento foram as de Kouylekov & Negri (2010), que apresentaram o EDITS[12], *Edit Distance Textual Entailment Suite*, algoritmo que verifica, justamente, o custo de edição entre duas sentenças, conseguindo aproximadamente 62% de acurácia no RTE-5. MacCartney (2009), por sua vez, apresentou o MANLI, um algoritmo de alinhamento também baseado no custo de edição. Esse algoritmo, fez parte do modelo apresentado na tese do autor, o Natlog, que obteve 59% de acurácia no RTE-3 e 70% no FraCas.

No outro grupo, o das metodologias que procuram representar o conteúdo semântico das sentenças dos pares, estão os modelos baseados em representações formais para o significado. Desde os anos iniciais do *Pascal RTE Challenge*, modelos baseados em formalizações foram amplamente explorados.

No RTE-1, os principais representantes dessa metodologia foram Bos & Markert (2005*a*), pesquisadores que apresentaram um modelo baseado na lógica de primeira ordem, obtendo uma acurácia de 55%. O trabalho dos autores contou ainda com uma discussão sobre a qualidade da classificação dos pares do corpus e sobre a classificação binária (verdadeiro-falso). Sobre a qualidade da classificação dos pares, os autores apontaram alguns dados aos quais julgaram que o acarretamento não existia e, portanto, deveriam ser classificados como falsos. Quanto à classificação binária, sugeriram que nos desafios posteriores a classificação "contraditório" fosse incluída.

Em trabalho posterior ao desafio, Bos & Markert (2005*b*) implementaram um modelo híbrido com algumas semelhanças ao que apresentaram no desafio. Nesse trabalho, porém, os autores procuraram discutir como abordagens baseadas em formalizações lógicas tinham a performance limitada pela necessidade de lidar com palavras que denotavam conhecimento de mundo. Os resultados, no entanto, foram melhores do que aqueles apresentados no desafio, 61% de acurácia no RTE-1.

[12] http://edits.fbk.eu/

Modelos baseados em formalizações lógicas oferecem a possibilidade de representar o conteúdo semântico de textos. No entanto, as características linguísticas dos RTEs comprometeram a performance desses modelos nos anos iniciais do desafio. As formalizações lógicas permitiam que os algoritmos conseguissem alta precisão na avaliação dos acarretamentos. A cobertura, porém, era limitada pela variabilidade linguística dos corpora.

Metodologias baseadas no aprendizado de máquina talvez tenham sido as mais utilizadas ao longo dos RTEs. Trata-se, quase sempre, de abordagens híbridas em que diferentes traços são extraídos do conjunto de dados para realizar a classificação. No RTE-1, temos trabalhos como o de Raina *et al.* (2005) e Newman *et al.* (2005), pesquisadores que apresentaram um algoritmo de árvore de decisão para classificar os pares, obtendo, ambos, uma acurácia de 56%.

Os modelos baseados em aprendizado de máquina foram, como afirmado por Dagan *et al.* (2013), os maiores representantes de arquiteturas híbridas. Ao longo dos desafios, esses modelos incorporaram os mais variados módulos de análise dos pares, como alinhamento, traços de similaridade e até mesmo a criação de conjuntos de dados específicos para treinamento, já que a base de dados dos RTEs muitas vezes não era suficiente para suprir as demandas de etapas como essa.

A proposta de MacCartney (2009) foi baseada na lógica natural, até então não utilizada nos desafios. O autor não participou do primeiro desafio com o modelo apresentado em sua tese, o NatLog. Porém, implementou um Bag-of-Words similar ao de Glickman *et al.* (2005) e o utilizou como *baseline* em três conjuntos de dados do *Pascal Challenge*: o RTE-1, o RTE-2 e o RTE-3. O valor de acurácia para o modelo ficou em 53%, 57% e 63%, respectivamente. Em seu trabalho, o autor procurou construir um modelo que desse conta de aspectos pragmáticos da língua, como no caso dos factivos: "Ed did not forget to force Dave to leave $\models$ Dave left" (MacCartney, 2009, p. 13).

O NatLog, resumidamente, é um sistema que verifica a presença de implicação por meio de uma etapa de alinhamento entre os pares e, posteriormente, por uma etapa de avaliação de acarretamento. Ocorre que, entre o alinhamento e a classificação final, diversas etapas são realizadas para extrair informações úteis a classificação. O sistema conta com análise linguística, com o alinhamento por

meio da edição dos pares e do cálculo do custo, com classificação de implicações no nível da palavra e com a combinação dessas informações em uma etapa final de avaliação de acarretamento. MacCartney (2009) não avalia seu modelo no RTE-1, mas sim no RTE-3 e no FraCas[13]. No RTE-3 o modelo obteve uma acurácia de 59% e uma precisão de 73%, enquanto no FraCas a acurácia foi 70% e a precisão foi 65%.

Os trabalhos brevemente descritos nesta seção nos oferecem uma ideia da dificuldade que a NLI representa para a comunidade científica. Com algumas exceções, como os trabalhos de Hickl *et al.* (2006), 75% de acurácia no RTE-2, e Hickl & Bensley (2007*b*), 80% no RTE-3, a média dos resultados ficou entre 50% e 60% de acurácia. Segundo Bentivogli *et al.* (2011), no RTE-7 já não havia, inclusive, grandes inovações por parte das metodologias apresentadas.

Modelos baseados em redes neurais artificiais praticamente não foram utilizados nos RTEs, ao menos na época dos desafios. Uma das razões era o tamanho da base de dados, como apresentamos anteriormente. O uso de modelos com arquiteturas desse tipo ganhou força com a criação de conjuntos de dados como o SNLI (Bowman *et al.*, 2015), corpus que ofereceu uma base de dados com tamanho suficiente para etapas de treino.

O surgimento de modelos baseados na transferência de aprendizado (Vaswani *et al.*, 2017), como o BERT, *Bidirectional Encoder Representations from Transformers* Devlin *et al.* (2018), estabeleceram outro padrão de comparação nas tarefas em NLP e, mais especificamente, na NLI. Treinados em grandes bases de dados e necessitando de etapas de *fine-tuning*, esses modelos estão estabelecendo o estado da arte nas mais variadas áreas da NLP.

No MultiNLI temos o trabalho de Liu *et al.* (2019*a*), que obteve 91% de acurácia. Modelos com arquiteturas similares também têm sido avaliados nos RTEs. Atualmente, o GLUE[14] apresenta uma série de conjuntos de dados. Entre esses conjuntos há uma junção dos RTEs 1, 2, 3 e 5. Liu *et al.* (2019*b*) obtiveram 86% de acurácia nessa nova versão do corpus, enquanto Devlin *et al.* (2018), como apresentado anteriormente, obtiveram uma acurácia de 70%.

[13] https://nlp.stanford.edu/~wcmac/downloads/

[14] https://gluebenchmark.com/

2.5 A NLI sob uma Perspectiva Crítica

Ao mesmo tempo que a NLI e o *Pascal RTE Challenge* foram ganhando atenção da comunidade científica, alguns trabalhos com análises teórico-qualitativas sobre os conjuntos de dados foram apresentados. Esse movimento de reflexão crítica acompanha a NLI até os dias de hoje. O título da seção, embora muito geral, pretende remeter a alguns dos principais trabalhos sobre os RTEs apresentados nessa linha.

Entre os avanços nos resultados, novos modelos avaliados e também novos conjuntos de dados, a NLI foi constantemente avaliada também em relação à sua definição, aos conjuntos de dados disponibilizados e ao seu escopo. Não pretendemos realizar um levantamento de todos os trabalhos que se enquadram, de uma maneira ou de outra, entre as críticas, mas apenas trazer aqueles mais comumente citados na literatura sobre o tema.

A primeira crítica foi apresentada por Zaenen *et al.* (2005) ainda no primeiro ano do desafio. Os pesquisadores apontaram para a necessidade de delimitar de maneira mais precisa o escopo dos acarretamentos que deveriam ser avaliados. Mesmo reconhecendo a importância e os esforços dos proponentes da tarefa, advertiram sobre as dificuldades de automatizar o conhecimento de mundo. Delimitar, de maneira rigorosa, o conhecimento linguístico necessário para realizar satisfatoriamente a tarefa seria o melhor caminho para oferecer à comunidade científica aquilo pelo qual de fato as máquinas deveriam ser avaliadas: seu entendimento textual.

Na realidade, até mesmo o nome da tarefa deveria ser modificado, segundo esses pesquisadores. Para eles, seria mais adequado nomeá-la como Reconhecimento de Inferência Textual. A NLI seria mais bem delimitada, então, se apresentasse em seus conjuntos de dados distinções entre diferentes tipos de inferências textuais necessárias para a classificação. Entre os tipos de inferências, os pesquisadores distinguiram pelo menos três: acarretamentos, implicaturas convencionais e implicaturas conversacionais.

Os acarretamentos, segundo Zaenen *et al.* (2005), seriam os casos menos controversos de inferência textual, se definidos nos moldes que apresentamos na Seção 1 do primeiro capítulo, isto é, em uma acepção lógica. Isso porque

sua classificação estaria restrita ao que foi declarado no nível das sentenças dos pares.

As implicaturas convencionais, apresentadas em Grice (1975) como aquelas que dependem das expressões linguísticas, precisariam ser distinguidas, de acordo com Zaenen *et al.* (2005), porque são baseadas em informações supostamente verdadeiras. Nesse sentido, não fariam, necessariamente, parte do que torna uma sentença verdadeira. Esse tipo de implicatura seria importante principalmente em pares identificados com IE. Em (1), exemplo retirado de Zaenen *et al.* (2005), temos uma sentença com uma implicatura convencional.

(1) The New York Times reported that Hanssen, who sold FBI secrets to the Russians, could face the death penalty.

O exemplo (1) apresenta um tipo de oração frequentemente explorado nos conjuntos de dados do *Pascal RTE Challenge*, a saber, as orações apositivas. Segundo Zaenen *et al.* (2005), a informação da oração apositiva do exemplo (1) pressupõe um conhecimento prévio de que Hanssen tenha vendido segredos do FBI para os russos. Esse conhecimento é diferente da nova informação veiculada pela oração principal do período, isto é, a de que o New York Times publicou a notícia de que Hanssen pode pegar a pena de morte.

Em alguns pares dos RTEs, o acarretamento está relacionado com a oração apositiva da premissa. Pares com essa configuração deveriam, de acordo com Zaenen *et al.* (2005), ser distinguidos quanto ao tipo de informação necessária para avaliar o acarretamento.

Por fim, a distinção entre implicaturas conversacionais, de acordo com Grice (1975), aquelas que suscitam informações contextuais, também deveria ser feita. Por pressupor informações contextuais, esse tipo de implicatura exigiria algo para o qual os modelos não estariam, necessariamente, preparados. Além disso, segundo Zaenen *et al.* (2005), o RTE não delimitava esse tipo de informação de maneira clara em seus dados. Um exemplo fornecido pelos autores pode ser observado em (2).

(2) He certainly has three children.

Na análise de Zaenen *et al.* (2005), ao proferir (2), um falante se compromete com a informação que está veiculando. Consequentemente, na ausência de evidências contrárias, tendemos a concluir que o indivíduo a quem o pronome *He* se refere não tem mais do que três filhos. No entanto, a sentença que desencadeia esse tipo implicatura poderia ser cancelada por outra como "In fact he has five, three daughters and two sons". Casos similares a esse, para Zaenen *et al.* (2005), também deveriam ser distinguidos nos desafios do *Pascal RTE Challenge*.

A proposta dos pesquisadores foi na direção de um melhor refinamento do conjunto de dados do RTE e também em sua metodologia de anotação. Mesmo sabendo da dificuldade em separar conhecimento linguístico de conhecimento de mundo, uma melhor delimitação dos tipos de inferências seria, segundo os autores, o caminho para uma maior consistência por partes dos anotadores, que em alguns casos não deixaram claro se a relação entre um par pressuporia conhecimento de mundo ou não. Além disso, pesquisadores sugeriram que os pares fossem extraídos de ferramentas mais diversificadas, evitando a similaridade lexical, característica de pares identificados com as tarefas CD.

Em resposta a Zaenen *et al.* (2005), Manning (2006) argumenta que delimitar tipos de inferência, sobretudo tipos de inferência comuns a lógicos e linguistas, poderia excluir inferências realizadas de maneira corriqueira por pessoas sem esse tipo especializado de conhecimento. De acordo com o autor, avaliar inferências feitas por humanos teria importância prática para sistemas de NLP, que muitas vezes lidam com tarefas de domínio aberto.

Sendo assim, Manning (2006) discorda de Zaenen *et al.* (2005) no que diz respeito às delimitações sugeridas por estes. Para o autor, a própria dificuldade em delimitar o conhecimento de mundo (ou o senso comum) de uma pessoa já tornaria difícil delinear o tipo de conhecimento necessário para classificar pares.

Assim, ao definirem a NLI do modo como fizeram, tendo como horizonte outras tarefas de NLP nas quais seria possível obter um padrão de julgamento humano, os criadores do *Pascal RTE Challenge* fizeram uma boa escolha, segundo Manning (2006). Principalmente por destacarem que o conhecimento de mundo não deveria ser o único parâmetro para julgar um acarretamento, mas também o conhecimento linguístico.

Mais precisamente, em relação à distinção de tipos de inferências proposta por Zaenen *et al.* (2005), Manning (2006) contra-argumenta afirmando que excluir implicaturas conversacionais, sobretudo as particulares, seria excluir muito do que pessoas comuns entendem por significado, muitas vezes distante das noções formais.

Em resumo, mesmo considerando que a tarefa tem uma definição imprecisa, a NLI oferece, de acordo com o autor, a possibilidade de avaliar as mais variadas metodologias, encorajando o desenvolvimento da área. Mesmo a dificuldade em estimar o conhecimento de mundo necessário para julgar um par não seria um problema, haja vista a concordância dos anotadores no RTE-1, aproximadamente 80%, o equivalente a 0,6 em escala Kappa McHugh (2012).

Por fim, embora reconhecendo os acertos metodológicos dos criadores da NLI, Manning (2006) sugere que as premissas dos pares sejam compostas por parágrafos e não somente por uma única sentença. Com isso, os modelos testados teriam maiores contextos para avaliar um acarretamento e a comunidade científica também confiaria mais na qualidade do corpus.

A tréplica veio em Crouch *et al.* (2006) com afirmação de que Manning (2006) teria interpretado erroneamente o artigo de 2005, entendendo-o como uma proposta de exclusão de alguns tipos de inferências. De acordo com Crouch *et al.* (2006), sua proposta era, ainda no primeiro artigo, a de que uma melhor delimitação sobre a informação na qual o acarretamento estaria baseado traria mais qualidade ao conjunto de dados. O desafio seria, portanto, delimitar se os acarretamentos estavam baseados em informações linguísticas ou no conhecimento de mundo.

Como exemplo de diretivas de anotação, Crouch *et al.* (2006) apresentaram brevemente as determinações do KBEVAL[15], supostamente mais claras do que as do RTE-1. No KBEVAL, acarretamentos deveriam ser anotados quanto a três tipos de informações: (i) Polaridade, isto é, verdadeiro-falso-neutro, (ii) Força, isto é, o tipo de acarretamento, e (iii) a fonte, isto é, se linguística ou baseada no conhecimento de mundo.

[15] https://nlp.stanford.edu/projects/infer/KBEvalReannoation.html

As questões levantadas por Crouch *et al.* (2006) vão na direção de outros trabalhos sobre o RTE-1 e sobre os demais conjuntos de dados para NLI. A dificuldade em isolar fenômenos necessários à classificação penaliza as metodologias, o que dificulta estimar o quão eficazes poderiam ser. Com um corpus mais bem delimitado, seria mais fácil estimar a eficácia de um modelo na tarefa. Um modelo baseado em representações formais, por exemplo, não seria penalizado por classificações erradas em pares que estivessem baseados em conhecimento de mundo.

Com trabalho centrado em uma proposta de reformulação da definição apresentada no primeiro ano de desafio, Korman *et al.* (2018) discutem em que medida a tarefa poderia incorporar noções de acarretamento presentes em discussões lógicas. Na verdade, os autores advertem para o fato de que o esforço por uma melhor definição da tarefa não acompanhou os esforços práticos para as resoluções dos problemas. Uma melhor definição ajudaria, então, a desenvolver instruções mais claras para a anotação dos pares e, consequentemente, conjuntos de dados com maior qualidade.

Para apresentar uma proposta de definição para tarefa Korman *et al.* (2018) fazem um breve percurso por diferentes tratamentos sobre acarretamentos na literatura lógica e linguística. Entre as diferentes noções de acarretamento, os autores analisam a implicação material, o acarretamento lógico, noções de acarretamento baseadas na teoria da relevância e na implicação doxática, concluindo que nenhuma delas seria a mais adequada para a finalidade da NLI.

Sendo assim, concordam que Dagan *et al.* (2005) fizeram uma boa escolha quando definiram a NLI com base no senso comum e em uma noção que admitisse a probabilidade de um acarretamento. Porém, alertam para o fato de que um refinamento da definição poderia evitar que os anotadores validassem acarretamentos não existentes. A sugestão dos autores foi acrescentar à definição a necessidade de que humanos, ao lerem um par, considerassem a proposição da hipótese, a proposição da premissa e, além disso, achassem razoável classificar a relação entre ambas como um acarretamento.

Os trabalhos brevemente apresentados nesta seção representam algumas das avaliações e críticas sob as quais o *Pascal RTE Challenge* foi submetido ao longo dos anos. Como é possível perceber, as críticas são dirigidas a diferentes

aspectos da tarefa: definição, qualidade dos conjuntos de dados, orientações e metodologia de anotação dos pares. O objetivo dessa breve exposição foi mostrar que a NLI, apesar de sua importância e de seus avanços, ainda permanece como um tópico aberto de pesquisa, sendo constantemente reavaliada.

2.6 Bases de Conhecimento Externo para NLI

Ao longo dos desafio, boa parte dos participantes utilizaram técnicas para lidar com o conhecimento externo exigido nos pares do *Pascal RTE Challenge*. Os recursos utilizados pelos modelos quase sempre foram desenvolvidos para outras tarefas de NLP que não a avaliação de acarretamentos (Dagan *et al.*, 2013). Nesta seção, apresentamos alguns dos principais recursos utilizados em NLI e, principalmente, no *Pascal RTE Challenge*.

2.6.1 A WordNet

A WordNet (Miller, 1995), amplamente utilizada em NLP, foi um dos recursos mais explorados nos desafios. Foi desenvolvida para o inglês por pesquisadores da Universidade de Princeton. A WordNet especifica relações lexicais entre palavras em uma estrutura hierárquica em árvore. Por meio dela, é possível obter sinônimos, hipônimos, hiperônimos e outras relações lexicais para um determinado item.

Os nós das árvores são chamados de *Synsets*, conjuntos de sinônimos. Os sinônimos são as relações mais básicas da WordNet. Esse recurso continua em franca expansão, sendo aprimorado a cada ano. Atualmente, a WordNet possui 155.327 palavras e 175.979 *Synsets*. Há também uma versão em desenvolvimento para o português, o projeto OpenWordNet-PT (Paiva *et al.*, 2012).

O que fez da WordNet um recurso muito utilizado nos desafios foi a oportunidade de avaliar diferentes possibilidades de implicação entre significados no nível das palavras, como relações de hiperonímia. De acordo com Dagan *et al.* (2013), mesmo sendo um recurso popular, não há um consenso sobre qual das relações oferecidas deveria ser explorada nos desafios. Além disso, as limitações

do recurso para lidar com predicados fazem com que as análises fiquem restritas somente ao nível das palavras.

2.6.2 O FrameNet

O FrameNet (Baker *et al.*, 1998) é um recurso lexical que traz informações sobre *frames* semânticos de diferentes itens. Desenvolvido também para o inglês, esse recurso apresenta informações sobre predicados e argumentos de um determinado evento, informações muito úteis para a avaliação de acarretamentos. Por meio do FrameNet, é possível identificar acarretamentos entre predicados e argumentos de um mesmo *frame* ou de *frames* diferentes.

Esse recurso não foi tão popular quanto a WordNet durante os desafios. Mesmo assim, foi bastante explorado por diferentes metodologias. Ambos os recursos foram, inclusive, combinados para potencializar a performance dos modelos e minimizar problemas decorrentes de itens que denotavam conhecimento de mundo nos pares.

2.6.3 O VerbOcean

O VerbOcean (Chklovski & Pantel, 2004) é um recurso com aproximadamente 29 mil verbos interligados por ao menos quatro tipos de relações: similaridade, antonímia, força, relações temporais e o que os autores chamam de *enablement*, ou "capacitação". O objetivo dos autores era oferecer um recurso com informações verbais precisas e úteis para variadas áreas da NLP, como QA e MT.

A similaridade indica verbos com significados parecidos, mas não necessariamente sinônimos. A antonímia, como o próprio nome sugere, indica verbos com sentidos opostos. A força indica verbos similares, porém, salientando diferenças entre sentidos mais e menos "intensos". As relações temporais indicam semelhanças e diferenças temporais entre eventos denotados pelos verbos. Por fim, o *enablement*, ou capacitação, indica relações de causalidade entre verbos e também situações em que um verbo pressupõe a realização de um evento denotado por outro.

A possibilidade de analisar relações entre verbos nos pares fez com que o VerbOcean também fosse um recurso explorado ao longo dos desafios do *Pascal*

RTE Challenge. Diferente da WordNet, as informações sobre os verbos não são apresentadas em estruturas hierárquicas. No entanto, apesar de apresentar relações semânticas de forma independente, o VerbOcean contém informações refinadas sobre relações verbais, o que faz dele um importante recurso para a NLI.

2.6.4 O CatVar

O CatVar, *Categorial Variation*, (Habash & Dorr, 2003) é um recurso para o inglês que contém informações sobre variações de diferentes classes gramaticais para uma determinada palavra. No total, o CatVar possui aproximadamente 62 mil *clusters* de palavras com 96 mil lexemas únicos. Nos desafios, esse recurso foi utilizado quase sempre para potencializar análises feitas com a WordNet. A possibilidade de identificar as possíveis variações categoriais para uma palavra e uma boa cobertura de itens lexicais fizeram do CatVar um recurso frequentemente explorado nos desafios.

3 Objetivos

Nosso objetivo central é comparar e discutir diferentes métodos para solução de problemas em NLI. Modelos baseados em arquiteturas *Transformer* vêm alcançando resultados excelentes nas mais variadas tarefas em NLP. Porém, são pouco transparentes quanto aos critérios, aspectos linguísticos ou atributos importantes para a classificação (Clark *et al.*, 2019*b*). Modelos baseados em regras, por outro lado, possibilitam ao pesquisador ajustar atributos e parâmetros orientados por sua intuição, embora nem sempre alcancem a performance das arquiteturas *Transformer*. Assim, nossa intenção é investigar em que medida seria possível gerar classificações baseadas em regras de diferentes tipos que, ao mesmo tempo que fossem transparentes, pudessem concorrer minimamente com a performance observada nas arquiteturas *Transformer*.

Para tanto, vamos testar quatro métodos diferentes de classificação para os problemas em NLI oferecidos pelos três primeiros conjuntos de dados do *Pascal RTE Challenge*. Os métodos consistem em regras de Bag-of-Words sem alinhamento, regras baseadas em alinhamento sentencial, representação lógica para os textos, classificação probabilística com um Classificador Bayesiano Ingênuo e um método baseado na tarefa de QA implementado com um modelo de arquitetura *Trasnformer*, a saber, o RoBERTa (Liu *et al.*, 2019*b*).

Pontualmente, os objetivos específicos da pesquisa são:

- Avaliar o quanto os métodos cobrem da classificação humana para os conjuntos de dados;
- Comparar os métodos entre si;

- Comparar os resultados com trabalhos avaliados nos mesmos conjuntos de dados.

Como vimos no capítulo anterior, propostas de solução baseadas nesses mesmos métodos estiveram presentes ao longo da história da NLI. A novidade em nosso trabalho talvez seja basear um dos métodos na tarefa de QA. Esperamos que, tomada em seu conjunto, as comparações e discussão possibilitem estimar o quanto das soluções e das dificuldades enfrentadas residem na própria escolha dos métodos.

4 Metodologia

Neste capítulo, apresentamos a proposta metodológica deste trabalho. Inicialmente, detalhamos os conjuntos de dados selecionados para a pesquisa. Na segunda parte, descrevemos a etapa de pré-processamento dos conjuntos de dados. Em seguida, detalhamos os pseudocódigos dos métodos testados, a saber, os métodos de Bag-of-Words (BoW) sem alinhamento, os métodos com alinhamento, o método de representação lógica e, por fim, o teste baseado na tarefa de QA com RoBERTa (Liu *et al.*, 2019*b*).

Os algoritmos e modelos foram implementados[1] em linguagem Python (versão 3.6.12) e executados no ambiente Jupyter[2], com exceção dos testes com o RoBERTa, rodados no Google Colab[3] em função do acesso a um ambiente de execução configurado como GPU.

4.1 Os Conjuntos de Dados

Nesta seção, apresentamos os três conjuntos de dados selecionados em nossa pesquisa. Como mencionamos anteriormente, são o RTE-1, o RTE-2 e o RTE-3, dados dos três primeiros anos do *Pascal RTE Challenge*. Os conjuntos de dados

[1] Os scripts com a implementação dos algoritmos podem ser acessados por meio do seguinte repositório: https://github.com/Rodrigo-SSouza/Scripts-of-my-Master-s-dissertation

[2] https://jupyter.org/

[3] https://colab.research.google.com/notebooks/intro.ipynb

possuem poucas diferenças entre si, característica dos cinco primeiros desafios. Todos são disponibilizados abertamente em formato XML[4].

4.1.1 O RTE-1

O RTE-1, primeiro da série de desafios do *Pascal RTE Challenge*, foi o corpus disponibilizado por Dagan *et al.* (2005) ao criarem a tarefa de NLI, na época ainda chamada de RTE. Como apresentado na Seção 1.1, os autores pretendiam introduzir uma nova tarefa que generalizasse um problema presente em diferentes áreas de pesquisa em NLP: avaliar acarretamentos entre frases e itens lexicais, considerando a variabilidade semântica das expressões linguísticas.

Ao todo, o corpus possui 800 pares compostos por premissas e hipóteses. A classificação para a relação de acarretamento é binária (verdadeiro-falso) e balanceada, isto é, o corpus contém 50% dos pares classificados como verdadeiros e 50% como falsos. Outra característica que merece destaque é a identificação que cada par possui para diferentes tarefas de NLP. Como apresentamos anteriormente, os anotadores criaram os pares com base em tipos específicos de tarefas para as quais a identificação do acarretamento seria útil. São sete tipos de tarefas: QA, MT, IE, IR, CD, RC e PP. Essas tarefas são igualmente distribuídas entre pares classificados como verdadeiros e falsos para a relação de acarretamento.

De acordo com Bentivogli *et al.* (2017), o RTE-1 é uma exceção aos outros corpora do desafio em relação ao processo de anotação para classificação de acarretamento. Enquanto os demais corpora foram anotados por três especialistas com conhecimento em linguística, o RTE-1 foi anotado somente por dois anotadores. Os pares mais adiante representam uma pequena parcela da complexidade presente no corpus:

(1) ID:48, Classification: "TRUE", Task: PP
P: Clinton is a very charismatic person.
H: Clinton is articulate.

(2) ID: 1609, Classification: "TRUE", Task: IE
P: The recent G8 summit, held June 8-10, brought together leaders of the

[4] https://tac.nist.gov//data/RTE/index.html

world's major industrial democracies.
H: The recent G8 summit took place on June 8-10.

Em (1) não existe uma relação de acarretamento entre ser uma pessoa muito carismática e ser articulado, ou eloquente. Uma rápida consulta a um Tesauro[5] permite constatar que as duas palavras também não compartilham uma relação de sinonímia. A quantidade de matérias sobre o carisma e o discurso articulado do ex-presidente estadunidense é abundante em páginas da *Web*, o que permitiria associar carisma e eloquência à sua imagem. Isso, porém, não seria suficiente para atribuir a classificação "True" para o acarretamento. A ausência de acarretamento entre *charismatic* e *articulate* deveria, portanto, inviabilizar a classificação "TRUE" para o par.

O par do exemplo (2), por sua vez, apresenta outro tipo de dificuldade para a classificação automática: a alternância sintática. Além da associação entre *held* e *took place*, um sintagma que ocorre na oração apositiva da premissa é colocado na oração principal da hipótese.

Os exemplos possuem também uma categoria específica de palavras que fazem parte de outra área de pesquisa em NLP: as Entidades Nomeadas, palavras que denotam nomes próprios, localizações geográficas, organizações e datas (Nadeau & Sekine, 2007). Entidades nomeadas dificultam a classificação dos problemas de NLI, pois, na maioria das vezes, pressupõem conhecimento de mundo. Nos pares em questão, *Clinton* e *G8* são representantes desses itens.

No ano em que foi disponibilizado para o primeiro desafio, o RTE-1 chamou atenção da comunidade científica por sua complexidade. Entre os trabalhos submetidos na avaliação, esteve o de Vanderwende & Dolan (2005). Esses autores não propuseram um modelo para resolver os problemas oferecidos pelo corpus. Ao contrário disso, analisaram os pares para verificar quantos poderiam ser classificados com base apenas em informações sintáticas. A análise apresentada por por Vanderwende & Dolan (2005) mostrou que em aproximadamente 37% dos pares é possível avaliar acarretamentos somente com base em informações sintáticas. Por outro lado, com a ajuda de ferramentas como

[5] https://www.thesaurus.com/browse/charismatic?s=t

um Tesauro, esse número aumentaria, chegando a aproximadamente 49%. As porcentagens nos dão estimativas da complexidade do corpus.

Ao longo dos anos de pesquisas e aprimoramentos em NLI, os corpora do *Pascal RTE Challenge* foram avaliados em relação à sua qualidade e representatividade para o fenômeno do acarretamento. Zaenen *et al.* (2005), por exemplo, discutem a dificuldade em separar conhecimento de mundo de conhecimento linguístico no RTE-1 e argumentam que alguns pares não deveriam ser classificados como verdadeiros para a relação de acarretamento.

De acordo com Dagan *et al.* (2005), porém, os anotadores concordaram em aproximadamente 80% das anotações dos pares, o que equivale a 0,6 na medida Kappa. Os 20% restantes do processo de extração, sem concordância, foram descartados pelos criadores da tarefa. A acurácia obtida pelos modelos avaliados ficou entre 50% e 60%, valores consideravelmente abaixo dos 80% que representam a classificação humana.

O melhor resultado no RTE-1 foi obtido por Delmonte *et al.* (2005) com o VENSES (Venice Semantic Evaluation System). Baseado em métodos de penalidade e recompensa para a similaridade lexical, o VENSES atingiu uma acurácia de 60%. Na quinta edição do desafio, Bentivogli *et al.* (2009) apresentaram um *baseline* que foi avaliado também nas quatro edições anteriores. O modelo foi baseado na quantidade de palavras em comum entre os textos dos pares. As *stop words*, porém, foram removidas antes da contagem. No RTE-1, o modelo implementado obteve uma acurácia de 55%.

4.1.2 O RTE-2

O RTE-2, segundo conjunto de dados do desafio, foi apresentado por Bar-Haim *et al.* (2006). Os procedimentos de criação do corpus foram os mesmos do ano anterior, o que fez com que RTE-2 e o RTE-1 fossem bastante semelhantes. No total, o corpus tem 1.600 pares divididos igualmente entre conjunto de teste e conjunto de validação. A classificação verdadeiro-falso, assim como no primeiro ano, também é igualmente dividida.

Em relação aos tipos de tarefas que identificam cada par, Bar-Haim *et al.* (2006) decidiram focar em apenas quatro, QA, IE, SUM e IR, também igualmente

distribuídas para a classificação de acarretamento e para a quantidade de pares. A extração dos pares foi feita de modo similar ao ano anterior, sendo que alguns foram extraídos diretamente de textos jornalísticos da *Web*, enquanto outros foram extraídos de conjuntos de dados para tarefas específicas, como o CLEF, para QA e o ACE-2004 para IE. Adiante, apresentamos dois pares do corpus.

(3) ID:691, Classification: "TRUE", Task: SUM
P: In one of the latest attacks, a US soldier on patrol was killed by a single shot from a sniper in northern Baghdad, the military said yesterday.
H: A sniper killed a U.S. soldier on patrol in Baghdad with a single shot.

(4) ID: 266, Classification: "TRUE", Task: IR
P: Green tea consumption is associated with decreased risk of breast, pancreatic, colon, oesophageal, and lung cancers in humans.
H: Tea protects from some diseases.

O par apresentado em (3), classificado como verdadeiro, é particularmente difícil de classificar automaticamente. O acarretamento está relacionado à informação da oração parentética da premissa. Além disso, essa mesma oração aparece na voz passiva, enquanto na hipótese aparece na ativa.

No exemplo apresentado em (4), por sua vez, temos outro cenário complexo para a classificação automática. Evidentemente, a hipótese está relacionada às informações apresentadas na premissa, o que justifica a classificação do acarretamento. Porém, todas as doenças apresentadas na premissa estão relacionadas ao hiperônimo que aparece na hipótese, exigindo o uso de um recurso como a WordNet para esse tipo de avaliação, por exemplo.

O RTE-2, assim como o primeiro conjunto de dados do desafio, possui uma grande variedade de fenômenos linguísticos relacionados aos acarretamentos. Garoufi (2007) apresentou uma análise do conjunto de dados, delimitando alguns desses fenômenos. Entre os fenômenos delimitados, estiveram hiperônimos, sinônimos, modificadores verbais, adjetivais e adverbiais, entre outros. Mantendo as características do RTE-1, o corpus é composto por problemas difíceis de solucionar computacionalmente.

Durante a etapa de anotação dos pares, ao menos três anotadores deveriam concordar sobre a classificação. Os pares paras os quais não houve concordância foram descartados, assim como no RTE-1. De acordo com Bar-Haim *et al.* (2006), aproximadamente 18,2% dos pares criados foram descartados. Para o corpus final, isto é, aquele apresentado no desafio, a concordância entre os anotadores foi de 89%, equivalente a 0,78 em escala Kappa.

Os melhores resultados no segundo ano do desafio foram obtidos por Hickl *et al.* (2006) e por Tatu *et al.* (2006), trabalhos já mencionados anteriormente. Estes com uma acurácia de 73% e aqueles com uma acurácia de 75%. Hickl *et al.* (2006) apresentaram o GROUNDHOG System, um modelo híbrido com módulos de alinhamento, detecção de paráfrases e um classificador do tipo árvore de decisão. Como os organizadores do desafio não ofereciam um conjunto de dados com tamanho suficiente para treinar o modelo, os pesquisadores criaram um paralelo e similar ao RTE-2 para a etapa de treinamento. Tatu *et al.* (2006), por sua vez, apresentaram o COGEX, um modelo capaz de gerar representações lógicas para as sentenças, composto também por uma etapa de alinhamento e por uma etapa de avaliação de acarretamentos entre palavras por meio da WordNet. O *baseline* apresentado em Bentivogli *et al.* (2009), por sua vez, obteve uma acurácia de 54% no corpus.

4.1.3 O RTE-3

O RTE-3 (Giampiccolo *et al.*, 2007), terceiro conjunto de dados apresentado pelo *Pascal RTE Challenge*, seguiu a mesma estrutura dos dois anteriores. Como apresentamos na Seção 2.1 do Cap. 2, os proponentes do terceiro desafio pretendiam estimular a participação de novos pesquisadores, mas também possibilitar que "veteranos" avaliassem a evolução de seus modelos. O RTE-3 teve 1.600 pares divididos igualmente em conjunto de validação e conjunto de teste, assim como o RTE-2.

Em relação aos tipos de tarefa e às suas respectivas distribuições entre pares classificados como verdadeiros ou falsos para a relação de acarretamento, Giampiccolo *et al.* (2007) decidiram manter o que foi feito no ano anterior. Ao todo, são quatro tipos de tarefa, IR, IE, SUM e QA, identificando 200 pares cada.

Os procedimentos de criação dos pares também foram os mesmos adotados no RTE-2. Em (5) e (6), apresentamos dois exemplos do corpus.

(5) ID: 667, Classification: "TRUE", Task: SUM, Lenght: Short
P: A German nurse, Michaela Roeder, 31, was found guilty of six counts of manslaughter and mercy killing.
H: A German nurse was convicted of manslaughter and mercy killing.

(6) ID: 514, Classification: "TRUE", Task: QA, Lenght: Long
P: Among the conditions for knowing the Quran is knowing the language in which the Quran has been interpreted and understood. Our predecessors were intent on learning Arabic because of this, for as 'Amr Ibn Khattab said, "I prefer to travel for forty nights to interpret an Ayat in the Book of Allah rather than spend this time at the Masjid of the Prophet (PBU) fasting during the day and praying during the night."
H: Arabic is the language of the Quran.

Em (5) temos um par classificado como verdadeiro para a relação de acarretamento. Embora a premissa apresente informações que não ocorrem na hipótese, essas informações não eliminam o acarretamento. Faz sentido pensar que se a enfermeira, Michaela Roeder, foi declarada culpada, também foi condenada pelos seus atos.

Em (6), por sua vez, é possível depreender que o Alcorão foi escrito em árabe. No entanto, a premissa não apresenta essa informação de modo explícito, o que poderia dificultar a classificação automática.

Os exemplos apresentam, também, uma informação nova por relação aos pares dos anos anteriores do desafio, a saber, "Lenght", que se refere ao tamanho da premissa. Em (5) temos uma premissa maior do que a hipótese, característica dos conjuntos de dados do desafio, e uma identificação de "Short", informando que essa premissa é menor do que um grupo de outras premissas do corpus. Já em (6) temos a identificação "Long", indicando que aquela premissa é maior do que a média das premissas do corpus.

Essa foi uma das inovações apresentadas no RTE-3. Aproximadamente 17% do conjunto de teste é composto por pares nos quais a premissa é maior do que a

média dos anos anteriores. Com essa inovação, Giampiccolo *et al.* (2007) pretendiam oferecer pares mais desafiadores e com maior contexto para a classificação do acarretamento. Outra inovação apresentada pelos proponentes do desafio foi a criação de um ambiente virtual de compartilhamento. Com esse ambiente, Giampiccolo *et al.* (2007) esperavam estimular os participantes a compartilharem recursos e discussões ao longo do desafio.

O RTE-3 teve o mesmo procedimento de anotação do RTE-2. Três anotadores avaliaram os pares, sendo que aqueles sem concordância foram descartados. Aproximadamente 19,2% dos pares foram removidos após a anotação. Ao contrário dos dois conjuntos de dados anteriores, o RTE-3 apresenta uma leve diferença na distribuição entre pares verdadeiros e falsos. Tanto para o conjunto de validação quanto para o conjunto de teste, a distribuição ficou com a seguinte configuração: 51,50% de pares falsos e 51,25% de pares verdadeiros. A concordância para o corpus ficou em 87,8%, o equivalente a 0,75 na escala Kappa.

Assim como nos conjuntos de dados anteriores, a presença de pares que pressupõem conhecimento de mundo para a validação dos acarretamentos é frequente. Clark *et al.* (2007), a partir da análise de 100 pares classificados como verdadeiros, identificaram três tipos diferentes de conhecimento de mundo necessários para classificar os pares: geral, que se refere a fatos gerais sobre o mundo, temporal e espacial, que envolve raciocínio numérico, e conhecimento de *frames*, que envolve o conhecimento humano sobre estrutura de eventos.

O melhor resultado no RTE-3 foi obtido por Hickl & Bensley (2007*a*), com uma metodologia similar a do ano anterior. O GROUNDHOG System, modelo já avaliado no RTE-2, atingiu uma acurácia de 80% no RTE-3. O segundo lugar ficou com Tatu & Moldovan (2007), que também utilizaram o COGEX, mantendo uma acurácia próxima da atingida no ano anterior, a saber, 72%. Já o *baseline* de Bentivogli *et al.* (2009) alcançou uma acurácia de 62% no conjunto de dados.

O terceiro desafio trouxe também um projeto piloto. Nesse projeto, chamado *Extending the Evaluation of Inferences from Texts*, pretendia-se estimular os pesquisadores a participarem de uma tarefa opcional: distinguir os pares classificados como falsos em contraditórios ou neutros e, se possível, oferecer uma justificativa para distinção.

4.2 Pré-Processamento dos Dados

Os conjuntos de dados do Pascal RTE Challenge são disponibilizados em formato XML, o que faz deles dados estruturados. Realizamos, então, uma etapa de pré-processamento para extrair os pares e prepará-los para os testes pretendidos. Nos itens adiante, listamos as etapas de pré-processamento.

- **Limpeza dos dados:** a primeira etapa realizada foi a limpeza dos textos dos pares. Embora os dados dos RTEs sejam estruturados, a presença de caracteres especiais é frequente em seus textos, o que poderia influenciar os resultados dos testes realizados.

- **Tokenização:** nesta etapa, segmentamos as sentenças dos pares em *tokens*, palavras e dígitos, e convertemos as palavras em minúsculas. As pontuações não foram removidas, pois poderiam nos dar pistas da organização sintática dos textos. A segmentação foi realizada com base no espaço em branco entre os *tokens* e, nesse sentido, a noção de palavra empregada na *tokenização* não coincide com acepções discutidas na literatura linguística. Sabemos, no entanto, que embora espaços em branco sejam úteis do ponto de vista prático, nem sempre são os parâmetros mais adequados para a segmentação (Manning & Schutze, 1999, p. 125).

- **Pos-Tagging:** a etiquetagem morfossintática foi realizada com o spaCy[6], ferramenta largamente utilizada nas mais variadas tarefas de NLP. A etiquetagem morfossintática foi uma etapa útil, sobretudo para identificarmos entidades nomeadas e explorarmos as características sintáticas dos pares.

- **NER:** reconhecimento de entidades nomeadas com o spaCy. Essa etapa difere da anterior por considerar as classificações do spaCy para diferentes tipos de entidades nomeadas.

- **Mascaramento de Entidades Nomeadas:** nesta etapa, os *tokens* reconhecidos como entidades nomeadas pelo spaCy foram mascarados com *strings*

[6] https://spacy.io/

do tipo "Entidade-1", "Entidade-2" e, assim, sucessivamente. No processo, mascaramos primeiro as entidades nomeadas da premissa. Dessa forma, buscamos mascarar as entidades nomeadas da hipótese com base nas mesmas máscaras das entidades nomeadas das premissas, caso co-ocorressem, ou com máscaras diferentes, caso não co-ocorressem. A primeira entidade nomeada da premissa, portanto, foi mascarada como "Entidade-1". Caso essa entidade ocorresse na hipótese, independente de sua posição na sentença, receberia a mesma etiqueta. Com isso, pretendíamos notar possíveis variações na estrutura sintática dos pares. Nesta etapa, também verificamos a ocorrência de *substrings* que poderiam ser partes de entidades nomeadas entre premissas e hipóteses, como "Bill Clinton" e "Clinton".

- **Lematização:** redução das palavras dos pares às suas respectivas formas lemáticas, isto é, aos seus respectivos lemas.

- **Análise de Dependências:** realizamos a análise de dependências dos pares com o spaCy. Com essa etapa, pudemos identificar sujeitos, verbos, objetos e demais classes sintáticas para realizar testes de co-ocorrência nos pares.

- **Mascaramento de Sujeitos e Objetos:** esta etapa foi realizada de modo similar àquela com as entidades nomeadas. Desta vez, atribuímos máscaras do tipo "Sujeito-1", "Sujeito-2", "Objeto-1", "Objeto-2" etc. Mais uma vez, utilizamos a premissa como guia e buscamos mapear nas hipóteses as ocorrências de sujeitos e objetos. Caso esses itens co-ocorressem, atribuíamos a mesma máscara da premissa, caso não, atribuíamos uma máscara diferente. Assim como na etapa de mascaramento das entidades nomeadas, também verificamos a possível ocorrência de *substrings* entre sujeitos e objetos.

- **Eliminação das *stop words*:** A partir de uma lista composta por stopwords do spaCy, do NLTK[7] e do TextBlob[8], realizamos uma filtragem nos pares dos conjuntos de dados. Com a junção de diferentes listas, conseguimos um total de 403 *stop words*.

[7] https://www.nltk.org/

[8] https://textblob.readthedocs.io/en/dev/

- **Transcrição de dígitos por extenso:** Em alguns pares dos RTEs, há alternância entre dígitos e suas formas por extenso, por exemplo "8/eight". Para minimizar problemas decorrentes desse tipo de alternância e aumentar a eficácia de testes baseados em coincidência lexical, utilizamos o Num2Words[9]. Essa ferramenta permite converter dígitos em suas respectivas formas por extenso.

- **Remoção das Entidades Nomeadas:** após o teste de inclusão de entidades nomeadas, filtramos esses itens dos corpora. Esse procedimento foi realizado antes de calcular o valor de IDF para cada palavra.

- **Remoção de Dígitos:** Assim como no caso das entidades nomeadas, os dígitos foram filtrados após alguns dos testes com o BoW. Essa etapa foi utilizada antes do cálculo de IDF para as palavras dos pares.

- **Cálculo de IDF:** cálculo da frequência inversa de uma palavra em documentos (IDF). Nessa etapa, utilizamos o corpus SUBTLEX-us[10] (Brysbaert & New, 2009) para o cálculo, pois o tamanho reduzido dos três conjuntos de dados influenciaria no potencial da técnica.

- **Cálculo do Terceiro Quartil:** cálculo do terceiro quartil a partir do valor de IDF das palavras de cada par. O valor do terceiro quartil foi utilizado como limiar de significância para a atribuição de traços que alimentaram o Classificador Bayesiano Ingênuo.

Algumas etapas do pré-processamento foram utilizadas em estágios específicos do trabalho. Isso quer dizer que nem todas as etapas descritas anteriormente foram aplicadas antes das regras com o BoW, por exemplo. Para alguns dos nossos métodos, precisaríamos de informações que o pré-processamento poderia retirar. É o caso, por exemplo, dos dígitos. A conversão dos dígitos para suas formas por extenso inviabilizaria um dos nossos testes com os métodos do BoW.

[9] https://pypi.org/project/num2word/

[10] https://www.ugent.be/pp/experimentele-psychologie/en/research/documents/subtlexus

4.3 Métodos Sem Alinhamento

Nesta seção, apresentamos os pseudocódigos dos seis algoritmos implementados para classificar os conjuntos de dados. Os métodos foram baseados em regras de BoW sem alinhamento sentencial. Alguns foram utilizados para classificar pares falsos, outros para classificar pares verdadeiros.

4.3.1 Ocorrência de Dígitos Diferentes entre Premissa e Hipótese

Para esse método, implementamos um algoritmo para comparar dígitos em cada par, esperando que a ocorrência de dígitos diferentes na hipótese nos permitisse identificar pares com classificação falsa para o acarretamento. O teste ficou restrito apenas a dígitos, não comparando possíveis alternâncias com suas respectivas formas por extenso, como por exemplo "7" e "seven" e nem diferenças entre grandezas numéricas. Em 4.1 apresentamos o pseudocódigo do algoritmo implementado para testar o método.

Algoritmo 4.1: Dígitos Diferente na Hipótese

```
   Entrada: Par: ⟨P, H⟩
   Saída: Par classificado como False ou "Indefinido"
1  para cada ⟨P, H⟩ faça
2      pares_com_dígitos ← pares em que premissas e hipóteses têm dígitos
3      para cada par em pares_com_dígitos faça
4          dígitos_da_premissa ← lista de dígitos da P
5          dígitos_da_hipótese ← lista de dígitos da H
6          para cada dígito em dígitos_da_hipótese faça
7              se dígito não está em dígitos_da_premissa então
8                  return False
9              senão
10                 retorna "Indefinido"
11             fim
12         fim
13     fim
14 fim
```

No Algoritmo 4.1, uma função recebe um par e verifica se os textos, premissa e hipótese, possuem dígitos. Caso possuam, os pares são filtrados. Para cada par, os dígitos da premissa e da hipótese são adicionados às listas "dígitos_da_premissa" e "dígitos_da_hipótese", respectivamente. O último passo é checar se ao menos um dígito da hipótese não ocorre na premissa. O algoritmo retorna, então, a classificação *False*, caso essa condição seja satisfeita, do contrário retorna "Indefinido".

4.3.2 Presença de Dígitos na Premissa e Ausência na Hipótese

No teste em questão, implementamos um algoritmo para verificar pares em que premissas possuíssem dígitos e hipóteses não. Nossa expectativa era a de que a maioria desses pares fossem classificados como verdadeiros para o acarretamento, uma vez que, ao restringirmos o teste aos dígitos, não encontraríamos informações diferentes das premissas. O pseudocódigo do algoritmo implementado é apresentado em 4.2.

Algoritmo 4.2: Dígitos na Premissa

```
   Entrada: Par: ⟨P, H⟩
   Saída: Par classificado como True ou "Indefinido"
1  para cada ⟨P, H⟩ faça
2  |  premissa_com_dígitos ← dígitos de P
3  |  hipótese_com_dígitos ← dígitos de H
4  |  par ← ⟨premissa_com_dígitos, hipótese_com_dígitos⟩
5  |  para cada par faça
6  |  |  se hipótese_com_dígitos é vazia então
7  |  |  |  se premissa_com_dígitos não é vazia então
8  |  |  |  |  retorna True
9  |  |  |  senão
10 |  |  |  |  retorna "Indefinido"
11 |  |  |  fim
12 |  |  fim
13 |  fim
14 fim
```

O Algoritmo 4.2 recebe um par como entrada e adiciona a duas novas listas, "premissa_com_dígitos" e "hipótese_com_dígitos", os respectivos dígitos de cada

texto. As listas passam, então, a compor um novo par. Para cada premissa e hipótese dentro do novo par, o algoritmo verifica se a hipótese é vazia e se a premissa possui dígitos. Caso essa condição seja satisfeita, o algoritmo atribui a classificação *True*, do contrário classifica como "Indefinido".

4.3.3 Hipótese Maior que Premissa

Para classificar pares com hipóteses maiores do que premissas, implementamos o algoritmo cujo pseudocódigo é apresentado em 4.3. Nossa expectativa era de que pares com hipóteses maiores fossem, em sua maioria, falsos pelo fato de que essas hipóteses possivelmente estariam veiculando conteúdos novos por relação às premissas.

Algoritmo 4.3: Hipótese Maior que Premissa

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como False ou "Indefinido"

```
para ⟨P, H⟩ faça
    tamanho_da_premissa← quantidade de tokens em P
    tamanho_da_hipótese ← quantidade de tokens em H
    se tamanho_da_premissa < tamanho_da_hipótese então
        retorna False
    senão
        retorna "Indefinido"
    fim
fim
```

No Algoritmo 4.3, basicamente, uma função recebe um par com duas listas (premissa e hipótese) e verifica a quantidade de palavras em cada uma. Palavras duplicadas não são ignoradas. Caso a hipótese possua mais palavras do que a premissa, sendo, portanto, maior, o algoritmo retorna a classificação *False*, do contrário retorna "Indefinido".

4.3.4 Coincidência entre os Pares

Para avaliar o quanto a coincidência lexical poderia ser um bom método para classificar pares verdadeiros para a relação de acarretamento, utilizamos

um cálculo simples para obter um limiar de coincidência entre a premissa e a hipótese de cada par. Em 4.4 temos pseudocódigo do algoritmo.

Algoritmo 4.4: Coincidência Lexical entre os Pares

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como True ou "Indefinido"

```
para ⟨P, H⟩ faça
    tokens_da_premissa ← Conjunto(P)
    tokens_da_hipótese ← Conjunto(H)
    intersecção ← tokens_da_premissa ∩ tokens_da_hipótese
    união ← tokens_da_premissa ∪ tokens_da_hipótese
    qtde_inter ← quantidade de tokens na intersecção
    qtde_união ← quantidade de tokens na união
    coincidência ← qtde_inter / qtde_união
    limiar ← 0,6 para o RTE-1, 0,55 para o RTE-2 e 0,45 para o RTE-3
    se coincidência ≥ limiar então
        retorna True
    senão
        retorna "Indefinido"
    fim
fim
```

O Algoritmo 4.4 recebe como entrada um par de sentenças *tokenizadas*. As variáveis "tokens_da_premissa" e "tokens_da_hipótese" recebem, respectivamente, o conjunto de *tokens* de cada um dos textos do par. Para obter os conjuntos de *tokens* dos textos dos pares, utilizamos a função Conjunto. Verifica-se, então, a intersecção e a união entre os conjuntos. Após esse passo, o algoritmo computa as quantidades presentes na intersecção e na união para, por fim, estimar a coincidência. A coincidência é dada pela divisão da quantidade de *tokens* na intersecção entre os conjuntos pela quantidade de *tokens* na união. Por fim, estabelecemos um critério de coincidência que deveria servir como um limiar de separação entre os pares. Para o nosso teste, escolhemos o valor de 0,65 para o RTE-1, 0,55 para o RTE-2 e 0,45 para o RTE-3. Os pares com grau coincidência igual ou acima do limiar foram, então, filtrados e classificados como verdadeiros. Pares com grau de coincidência abaixo do limiar foram classificados como "Indefinidos".

4.3.5 Inclusão de Entidades Nomeadas

Entidades nomeadas são importantes para a avaliação de acarretamentos pelas razões que demonstramos brevemente ao longo do trabalho. Por serem informativas, é muito provável que se a premissa contiver entidades nomeadas a hipótese também terá. Sendo assim, decidimos testar um método que classifica pares com base na presença de ao menos uma entidade nomeada na hipótese ausente na premissa. Em 4.5 podemos observar o pseudocódigo do algoritmo implementado.

O Algoritmo 4.5 recebe um par e verifica se a premissa e a hipótese têm entidades nomeadas. Caso tenham, o par é adicionado à lista "Pares_Com_ENs". As entidades nomeadas da hipótese são, então, armazenadas na lista "en_hipótese" para serem comparadas com a premissa. O próximo passo executado pelo algoritmo é verificar se ao menos uma entidade nomeada da hipótese não está presente na premissa. O algoritmo deve retornar *False*, caso essa condição seja satisfeita, e "Indefinido", caso não seja. Nesse método testamos duas possibilidades de classificação. Na primeira utilizamos o algoritmo com o reconhecimento de entidades nomeadas sem a etapa de mascaramento. Na segunda, utilizamos o mascaramento.

4.3.6 Presença de Hipóteses Vazias

Com esse método, decidimos verificar quantos pares possuíam hipóteses vazias após o filtro das *stop words* e das entidades nomeadas, classificando-os como verdadeiros, uma vez que nenhuma informação nova era adicionada à premissa (Algoritmo 4.6).

O Algoritmo 4.6 recebe um par e verifica duas condições: se a premissa não é vazia e se a hipótese é vazia. Caso essa condição seja satisfeita, o algoritmo classifica o par como *True*, do contrário classifica como "Indefinido".

Algoritmo 4.5: Inclusão de Entidades Nomeada

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como False ou "Indefinido"

```
para cada ⟨P,H⟩ faça
    pares_com_ens ← pares em que P e H têm entidades nomeadas
    para cada par em pares_com_ens faça
        premissa ← P
        en_hipótese ← entidades nomeadas da hipótese
        para cada termo em en_hipótese faça
            se termo não está em P então
                retorna False
            senão
                retorna "Indefinido"
            fim
        fim
    fim
fim
```

4.4 Métodos com Alinhamento

Nesta seção, apresentamos os quatro métodos baseados em coincidência e alinhamento sentencial. Ao contrário dos anteriores, todos os métodos foram testados para classificar pares verdadeiros para a relação de acarretamento.

4.4.1 Coincidência entre Triplas

Para testar este método, extraímos triplas das premissas e hipóteses de cada par com o OpenIE (Angeli *et al.*, 2015). Triplas, de acordo com Wu & Weld (2010), são estruturas do tipo $\{\langle arg_1, rel, arg_2 \rangle\}$. Os argumentos, arg_1 e arg_2, são sintagmas nominais e *rel* é a relação semântica implícita entre os argumentos. O OpenIE extrai mais de uma tripla para uma única sentença. Para evitar triplas repetidas, realizamos uma etapa de refinamento que considerava, inclusive, análises de inclusão entre *strings* e *substrings*. Em 4.7 temos o pseudocódigo do algoritmo que implementamos para extrair as triplas.

Algoritmo 4.6: Par com Hipótese Vazia

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como True ou "Indefinido"

```
para cada ⟨P,H⟩ faça
    se P não é uma lista vazia então
        se H é uma lista vazia então
            retorna True
        senão
            retorna "Indefinido"
        fim
    fim
fim
```

Algoritmo 4.7: Extração de Triplas

Entrada: Par: $\langle P, H \rangle$
Saída: Triplas para cada par

```
para cada ⟨P,H⟩ faça
    TP ← lista de triplas de P extraídas com o OpenIE
    TH ← lista de triplas de H extraídas com o OpenIE
    triplas_de_p ← Conjunto(TP)
    triplas_de_h ← Conjunto(TH)
    LTP ← Lista(triplas_de_p)
    LTH ← Lista(triplas_de_h)
    retorna LTP, LTH
fim
```

No Algoritmo 4.7, o OpenIE extrai triplas para premissas e hipóteses de cada par fornecido como argumento. Após a extração, uma etapa de refinamento é realizada para retirar triplas repetidas para uma dada sentença de cada par. Essa etapa é realizada por meio da função Conjunto(), que retorna o conjunto de *tokens* de premissas e hipóteses. Na etapa seguinte, o algoritmo converte os conjuntos de triplas em listas de triplas, representadas pelas variáveis LTP, para premissas, e LTH, para hipóteses. Por fim, o algoritmo retorna a lista de triplas de premissas e hipóteses dos pares. No exemplo (7), retirado do RTE-3, temos o resultado obtido com a extração de triplas por meio do Algoritmo 4.7.

(7) ID: "126", Classification: "TRUE" Task: IE
P: André Henri Constant van Hasselt was born at Maastricht, in Limburg. He was educated in his native town, and at the university of Liège.
H: André Henri Constant van Hasselt was educated in Maastricht.
Triplas de P:
Tripla 1 – (André Henri Constant van Hasselt, was educated in, Maastricht)
Tripla 2 – (He, was educated in, his town)
Tripla 3 – (He, was, educated)
Triplas de H:
Tripla 1 – (André Henri Constant van Hasselt, was, educated)
Tripla 2 – (He, was, educated)

Em (7), temos um exemplo das triplas criadas para um par. Chamamos atenção principalmente para as triplas da hipótese. O algoritmo gerou uma tripla com o pronome *He*, mesmo que esse pronome não apareça no texto original.

O Algoritmo 4.7 foi utilizado como parte de outro algoritmo, a saber, um que compara o grau de coincidência entre as triplas de cada par. Em 4.8 apresentamos o pseudocódigo do algoritmo que compara o grau de coincidência entre as triplas.

A primeira etapa executada pelo Algoritmo 4.8 consiste na extração de triplas de cada par. Como vimos anteriormente, a extração de triplas é feita por meio do Algoritmo 4.7, que gera listas de triplas para premissas e hipóteses. Essas listas são armazenadas na variável "triplas_do_par". Após a extração das triplas, as variáveis "triplas_de_p" e "triplas_de_h" recebem os itens nas posições [0] e [1] de "triplas_do_par". Esses itens correspondem, respectivamente, às listas de triplas da premissa e hipótese de cada par. Além da extração das triplas, um contador, representado pela variável "total", é inicializado com o valor 0. Em nosso teste, o cálculo do grau de coincidência ficou restrito à triplas que possuem o argumento *rel* idêntico. Lembremos que as triplas são compostas por três tipos de informação: $\{\langle arg_1, rel, arg_2 \rangle\}$. Os argumentos arg_1 e arg_2 são sintagmas nominais e *rel* é uma relação semântica entre os argumentos. No Algoritmo 4.8, esse argumento é representado por "relação_de_p", para a premissa, e "relação_de_h", para a hipótese. Sendo assim, o algoritmo verifica, para

Algoritmo 4.8: Coincidência entre Triplas

```
   Entrada: Par ⟨P, H⟩
   Saída: Par classificado como True, "Indefinido" ou "Relações Diferentes"
 1 para cada ⟨P, H⟩ faça
 2     triplas_do_par ← Extração(⟨P, H⟩) [Algoritmo 4.7]
 3     triplas_de_p ← triplas_do_par[0]
 4     triplas_de_h ← triplas_do_par[1]
 5     total ← 0
 6     para cada tripla_de_p em triplas_de_p faça
 7         para relação_de_p em tripla_de_p faça
 8             para cada tripla_de_h em triplas_de_h faça
 9                 para relação_de_h em tripla_de_h faça
10                     se relação_de_p == relação_de_h então
11                         intersecção ← tripla_de_p ∩ tripla_de_h
12                         união ← tripla_de_p ∪ tripla_de_h
13                         qtde_inter ← quantidade de tokens na intersecção
14                         qtde_união ← quantidade de tokens na união
15                         coincidência ← qtde_inter / qtde_união
16                         total += coincidência
17                         se total ≥ 0.6 então
18                             retorna True
19                         senão
20                             retorna "Indefinido"
21                         fim
22                     senão
23                         retorna "Relações Diferentes"
24                     fim
25                 fim
26             fim
27         fim
28     fim
29 fim
```

uma dada tripla da premissa em "triplas_de_p", representada por "tripla_de_p", se essa tripla possui o argumento *rel*, representado por "relação_de_p", igual ao argumento *rel*, representado por "relação_de_h", para uma dada tripla em "triplas_de_h", representada por "tripla_de_h". Caso essa condição seja satisfeita, o algoritmo verifica a coincidência entre as triplas. Na etapa de verificação da coincidência, o algoritmo computa a quantidade de *tokens* presentes na intersecção e na união entre as triplas ("tripla_de_p" e "tripla_de_h") para estimar o grau de coincidência. A coincidência é dada pela divisão da quantidade de *tokens* na intersecção pela quantidade de *tokens* na união entre as triplas. Como

a lista de triplas da premissa pode ter mais de uma tripla com a relação coincidente com triplas da lista de triplas da hipótese, o Algoritmo 4.8 soma o grau de coincidência obtido a cada iteração, atribuindo a soma à variável "total". Por fim, um limiar é estabelecido para classificar pares como verdadeiros ou indefinidos. Para o algoritmo em questão, escolhemos um limiar de 0,6. Pares com grau total de coincidência igual ou acima desse limiar foram classificados como *True* e pares com grau de coincidência abaixo do limiar foram classificados como "Indefinido".

4.4.2 Inclusão de Triplas ou Duplas

Neste método, implementamos um algoritmo para verificar em quantos pares ao menos uma tripla ou dupla da hipótese ocorria no conjunto expandido de triplas e duplas da premissa. Desta vez, porém, construímos as triplas e duplas a partir da análise de dependências realizada com o spaCy. A criação das triplas e duplas ficou restrita à oração principal dos textos de cada par. As triplas foram compostas de duas formas: sujeito, verbo e objeto, que poderia ser direto ou indireto, e também sujeito, verbo intransitivo e adjunto. As duplas foram compostas por sujeitos e verbos intransitivos.

O conjunto de triplas e duplas da premissa foi expandido com a WordNet. Para sujeitos e objetos que não fossem nomes de pessoas, extraímos sinônimos, hiperônimos e holônimos. Para os verbos, extraímos relações de causalidade, como "kill" causa "die", hiperônimos e relações de acarretamento. As duplas também nos deram a possibilidade de verificar outro tipo de relação que gera acarretamentos, a saber, a elisão de adjuntos. Nos pares do RTE-1 em (8), (9) e (10), apresentamos alguns exemplos dessas estruturas obtidas por meio da expansão.

(8) ID: "933", Classification: "TRUE", Task: IR
P: Crude Oil Prices Slump.
H: Oil price drop.
Duplas P:
Dupla 1 – (oil, slump);
Dupla 2 – (oil, **drop down**);

Dupla 3 – (oil, **drop**);
Dupla de H
Dupla 1 – (oil, drop).

(9) ID: "1996", Classification: "TRUE", Task: PP
P: Iraqi militant abduct 2 Turks in Iraq.
H: Iraqi militants kidnap 2 Turks in Iraq.
Triplas P:
Tripla 1 – (militant, abduct, turks);
Tripla 2 – (militant, **snatch**, turks);
Tripla 3 – (militant, **kidnap**, turks);
Tripla 4 – (militant, abduct, 2 turks iraq);
Tripla 5 – (militant, **snatch**, 2 turks in iraq);
Tripla 6 – (militant, **kidnap**, 2 turks iraq);
Tripla H:
Tripla 1 – (militant, **kidnap**, turks).

(10) ID: "932", Classification: "TRUE", Task: IR
P: Oil prices drop to two-month low.
H: Oil prices drop.
Duplas e Triplas P:
Dupla 1 – (oil, drop);
Dupla 2 – (oil, **drop down**);
Dupla 3 – (oil, **slump**);
Tripla 1 – (oil, drop, two-month low);
Tripla 2 – (oil, **drop down**, two-month low);
Tripla 3 – (oil, **slump**, two-month low);
Dupla H:
Dupla 1 – (oil, drop).

Em (8) e em (9), as relações obtidas com a WordNet permitiram construir duplas e triplas idênticas às que ocorrem nas respectivas hipóteses. Já em (10), temos um outro tipo de relação que também gera acarretamento: a elisão de

adjuntos da premissa. Após a expansão, realizamos o teste com o algoritmo cujo pseudocódigo é apresentado em 4.9.

Algoritmo 4.9: Inclusão de Triplas ou Duplas

```
   Entrada: Par: ⟨P, H⟩
   Saída: Par classificado como True ou "Indefinido"
1  para cada ⟨P, H⟩ faça
2  |  conjunto_expandido_de_p ← triplas ou duplas expandidas de P
3  |  conjunto_de_h ← triplas ou duplas de H
4  |  para cada tripla ou dupla em conjunto_de_h faça
5  |  |  se tripla ou dupla ⊂ conjunto_expandido_de_p então
6  |  |  |  retorna True
7  |  |  senão
8  |  |  |  retorna "Indefinido"
9  |  |  fim
10 |  fim
11 fim
```

O Algoritmo 4.9 recebe um par como argumento e verifica em quantos pares ao menos uma tripla ou dupla da hipótese está contida no conjunto expandido da premissa. Caso essa condição seja satisfeita, o algoritmo retorna a classificação *True*, do contrário, retorna "Indefinido".

4.4.3 Co-ocorrência entre Sujeitos

Para verificar a possibilidade de que pares verdadeiros apresentem informações similares sobre sujeitos, decidimos implementar um algoritmo para verificar co-ocorrências desse tipo. Para extrair sujeitos, realizamos a análise de dependências com o spaCy e também utilizamos as máscaras atribuídas no pré-processamento. Em 4.10, apresentamos o pseudocódigo do algoritmo implementado.

O Algoritmo 4.10 toma como argumento um par de premissa-hipótese para o qual a análise de dependências já foi realizada. A partir disso, o algoritmo extrai *tokens* classificados como sujeitos e os compara. Caso os sujeitos da premissa sejam iguais aos sujeitos da hipótese, o algoritmo retorna a classificação *True*. Do contrário, retorna "Indefinido".

Algoritmo 4.10: Co-ocorrência entre Sujeitos

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como True ou "Indefinido"

```
para cada ⟨P,H⟩ faça
    sujeitos_da_premissa ← lista de tokens classificados como 'sujeito' em P
    sujeitos_da_hipótese ← lista de tokens classificados como 'sujeito' em H
    se sujeitos_da_premissa == sujeitos_da_hipótese então
        retorna True
    senão
        retorna "Indefinido"
    fim
fim
```

4.4.4 Co-ocorrência entre Objetos

Assim como no teste anterior, decidimos avaliar quantos pares verdadeiros possuíam co-ocorrências entre objetos. Sendo assim, implementamos um algoritmo similar ao 4.10 para verificar a co-ocorrência entre esses itens. O pseudocódigo do algoritmo implementado para o teste realizado é apresentado em 4.11.

O Algoritmo 4.11, de modo similar ao Algoritmo 4.10, toma um par com informações sobre dependências sintática como argumento. Em seguida, extrai objetos, diretos ou indiretos, e verifica se co-ocorrem. Caso essa condição seja satisfeita, o algoritmo retorna *True*. Do contrário, retorna "Indefinido".

4.5 Método Baseado em Representação Lógica

Neste método, procuramos representar premissas e hipóteses formalmente. Para tanto, utilizamos o DeepCCG (Yoshikawa *et al.*, 2017), *parser* para gramáticas categoriais combinatórias que possui interface para o CCG2Lambda (Martinez-Gómez *et al.*, 2016), um *parser* para conversão de sentenças em forma lógica. Para nossos testes, adaptamos o CCG2Lambda de modo a obter os *outputs* que necessitávamos. O CCG2Lambda[11] fornece *outputs* preparados para etapas de

[11] https://github.com/mynlp/ccg2lambda

Algoritmo 4.11: Co-ocorrência entre Objetos

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como True ou "Indefinido"

```
para cada ⟨P,H⟩ faça
    objetos_da_premissa ← lista de tokens classificados como 'objeto' em P
    objetos_da_hipótese ← lista de tokens classificados como 'objeto' em H
    se objetos_da_premissa == objetos_da_hipótese então
        retorna True
    senão
        retorna "Indefinido"
    fim
fim
```

prova de teorema. Em seu trabalho, Martinez-Gómez *et al.* (2016) utilizaram o Coq[12] para essa etapa. Não pretendíamos, no entanto, utilizar uma etapa similar em nosso teste. Sendo assim, adaptamos o *parser* e criamos uma função para oferecer melhor legibilidade nos *outputs*. Os pares em (11), (12) e (13), retirados do RTE-1, RTE-2 e RTe-3, apresentam as formas lógicas geradas pelo *parser* com as adaptações que fizemos.

(11) ID: "828", Classification: "FALSE", Task: CD
P: Jennifer Hawkins is the 21-year-old beauty queen from Australia.
H: Jennifer Hawkins is Australia's 20-year-old beauty queen.
Forma Lógica de P:
$\exists x_1$ (hawkins(x_1) & jennifer(x_1) $\exists x_2$ (queen(x_2) & beauty(x_2) & 21 year old(x_2) & $\exists x_3$ (australia(x_3) & $\exists e_1$ (from(e_1, x_3) & (subj(e_1) = x_2))) & ($x_1 = x_2$)))
Forma Lógica de H:
$\exists x_1$ (hawkins(x_1) & jennifer(x_1) $\exists x_2$ (queen(x_2) & beauty(x_2) & 20 year old(x_2) & australia's(x_2) & ($x_1 = x_2$)))

(12) ID: "400", Classification: "TRUE", Task: QA
P: Leo Fender invented the first electric guitar and the electric bass guitar.

[12] https://coq.inria.fr/

H: Leo Fender invented the first electric guitar.
Forma Lógica de P:
$\exists x_1$ (fender(x_1) & leo(x_1) $\exists x_2$ (guitar(x_2) & eletric(x_2) & first(x_2) & $\exists e_1$ (invent(e_1) & (subj(e_1) = x_1) & (acc(e_1) = x_2))) & $\exists x_3$ (guitar(x_3) & bass(x_3) & electric(x_3) & $\exists e_2$ (invent(e_2) & (subj(e_2) = x_1) & (acc(e_2) = x_3))))
Forma Lógica de H:
$\exists x_1$ (fender(x_1) & leo(x_1) $\exists x_2$ (guitar(x_2) & eletric(x_2) & first(x_2) & $\exists e_1$ (invent(e_1) & (subj(e_1) = x_1) & (acc(e_1) = x_2))))

(13) ID: "489", Classification: "TRUE", Task: QA Lenght: Short sho
P: The world famous Gurkhas come from the high mountains of Nepal.
H: The Gurkhas come from Nepal.
Forma Lógica de P:
$\exists x_1$ (gurkhas(x_1) & famous(x_1) & world(x_1) & $\exists e_1$ (come(e_1) & (subj(e_1) = x_1) & $\exists x_2$ (mountain(x_2) & high(x_2) & $\exists x_3$ (nepal(x_3) & (x_2 = x_3)) & from(e_1, x_2))))
Forma Lógica de H:
$\exists x1$ (gurkhas(x_1) & $\exists e_1$ (come(e_1) & (subj(e_1) = x_1) & $\exists x_2$ (nepal(x_2) & from(e_1, x_2))))

Após a conversão dos pares em forma lógica, implementamos o algoritmo cujo pseudocódigo é apresentado em 4.12. Nosso teste consistiu em verificar a coincidência entre fórmulas, ou proposições, de premissas e hipóteses.

A primeira etapa executada pelo Algoritmo 4.12 é segmentar premissas e hipóteses em fórmulas. Após a segmentação, o valor 1 é atribuído a cada item que compõe uma determinada fórmula da premissa e armazenado na variável "total". Para cada fórmula da hipótese idêntica a uma determinada fórmula da premissa, a variável "coincidência" recebe o valor 1, caso contrário, recebe o valor 0. O algoritmo, então, calcula a proporção de coincidência entre os pares por meio da divisão do valor armazenado em "coincidência" pelo valor armazenado em "total". Para a classificação, o algoritmo retorna *True*, caso o limiar seja igual ou maior do que 0,60, do contrário, retorna "Indefinido".

4.6 O Classificador Bayesiano Ingênuo

Para classificar os pares dos conjuntos de dados por meio de técnicas de aprendizado de máquina supervisionado, implementamos um Classificador Bayesiano Ingênuo. Os parâmetros utilizados para selecionar os atributos para o treino do classificador são listados adiante:

- **Coincidência:** o Algoritmo 4.4 possibilitou estimar um valor de coincidência entre os pares. Ao verificar a variabilidade entre os valores, bem como a natureza desse tipo de variável, isto é, quantitativa, optamos por utilizá-los para selecionar traços de natureza qualitativa. O primeiro passo foi dividir o corpus ao meio, utilizando o cálculo da mediana para os valores de coincidência. Entendemos, porém, que a divisão em duas partes iguais não influenciaria o classificador. No segundo e último passo, escolhemos um valor acima da mediana, especificamente 0,26 para o RTE-1, 0,18 para o RTE-2 e 0,11 para o RTE-3, e o utilizamos como limiar para selecionar dois atributos: coincidência baixa ou não entre os pares.

- **Tipo de Tarefa:** os conjuntos de dados possuem identificação para tipos de tarefa de NLP (Compreensão de Leitura, Extração de Informação, Perguntas & Respostas..., como vimos no Cap. 2). Decidimos, então, utilizar os tipos de tarefa como atributos em alguns de nossos testes.

- **Palavras Relevantes:** uma das etapas do pré-processamento consiste em obter o valor informacional das palavras dos pares por meio do cálculo do IDF. Para selecionar traços a partir desse valor, calculamos o terceiro quartil em cada par. Com o valor do terceiro quartil foi possível estabelecer um limiar de significância e filtrar palavras com IDF abaixo desse limiar. O último passo foi comparar a quantidade de palavras relevantes nos textos. Com isso, teríamos três possibilidades de traços: mesma quantidade de palavras relevantes entre premissa e hipótese, premissa com mais palavras relevantes ou hipótese com mais palavras relevantes.

Os atributos selecionados para treinar o classificador foram, portanto, a comparação entre o total de palavras relevantes em cada texto do par, que

poderia ser maior na premissa, maior na hipótese ou igual, a coincidência baixa ou não e, para alguns testes, o tipo de tarefa. Para fixar cada um dos parâmetros necessários ao treinamento, atribuímos probabilidades à ocorrência dos atributos dentro das classes e também a cada uma das classes.

A equação do Classificador Bayesiano Ingênuo apresentada na Subseção 1.1.1 demonstra que, para estimar a probabilidade de que um documento pertença a uma classe, é necessário multiplicar probabilidades, o que pode resultar em valores muito baixos. Para evitar possíveis problemas decorrentes de valores numéricos baixos, utilizamos logaritmos. A fórmula em 4.1 é subjacente ao classificador implementado:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i=1}^{n} P(f_i|c) \tag{4.1}$$

Ao adicionar logaritmos, conseguimos evitar valores muito baixos para as probabilidades, pois passamos a lidar não com multiplicações, mas com somas. Basta então, somar os valores dos produtos convertidos para obtermos valores de estimativa para as classificações, como indicado pelo somatório $\sum$. Uma vez decidido cada um desses parâmetros, implementamos o classificador. A implementação realizada neste trabalho é semelhante àquela apresentada em Ferreira & Lopes (2019).

4.6.1 Treinamento

Para treinar o classificador, separamos aleatoriamente 80% dos pares de cada conjunto de dados. Os 20% restantes foram utilizados para teste . Calculamos, então, a probabilidade de cada classe, "TRUE" ou "FALSE", por meio da MLE, estimativa por máxima verossimilhança. As probabilidades foram obtidas com o cálculo simples da frequência relativa de cada classe presente no corpus:

$$\hat{P}(c) = \frac{N_c}{N_{doc}} \tag{4.2}$$

Para o nosso problema, estimamos a probabilidade de uma dada classe dividindo a quantidade de vezes em que esta classe ocorre pelo total de pares.

As probabilidades condicionais dos traços por relação às classes também foram obtidas por meio do cálculo da frequência relativa:

$$\hat{P}(f_i|c) = \frac{contagem(f_i, c)}{contagem(c)} \tag{4.3}$$

Para um dado traço, por exemplo, "premissa com mais palavras relevantes", calculamos sua probabilidade condicional por meio da contagem do número de vezes em que esse traço ocorre em cada classe, dividindo pelo número total de cada classe. No entanto, para evitar problemas decorrentes da possível ausência de algum dos atributos em uma das classes durante o treinamento e, consequentemente, uma probabilidade de valor zero, utilizamos a suavização de Laplace. As probabilidades condicionais foram obtidas, então, pela formula:

$$\hat{P}(f_i|c) = \frac{contagem(f_i, c) + 1}{contagem(c) + V} \tag{4.4}$$

Ao utilizar a suavização de Laplace procuramos evitar problemas que poderiam ocorrer caso um traço do corpus de teste não houvesse ocorrido no treino. A intuição por trás do procedimento é adicionar uma pequena probabilidade aos atributos que não ocorrem no treino e diminuir um pouco a probabilidade dos que ocorreram. O valor utilizado para a suavização na implementação do classificador deste projeto foi 1, tal como apresentado na fórmula. A variável V se refere ao tamanho do conjunto do vocabulário, isto é, palavras distintas.

4.7 Modelo Final

Como veremos no próximo capítulo, com exceção do método baseado em representação lógica, cada um dos algoritmos descritos até aqui fez parte de dois modelos finais avaliados nos conjuntos de dados. Os modelos foram compostos, basicamente, por três estágios: Pré-processamento, Regras de Classificação (ver seção 5.4) e um Classificador Bayesiano Ingênuo. O pseudocódigo em 4.13 apresenta o algoritmo do modelo e a ordem de aplicação de cada etapa.

Como o pseudocódigo do Algoritmo 4.13 mostra, após o Pré-processamento, outro algoritmo, denominado Regras de Classificação, é aplicado aos pares.

Esse algoritmo é composto pelos métodos de classificação descritos nas seções anteriores, exceto o método de representação lógica. A aplicação dos métodos que compõem o algoritmo Regras de Classificação segue os valores de acurácia em ordem decrescente obtidos por testes individuais em cada conjunto de dados. Por fim, o Classificador Bayesiano Ingênuo é aplicado nos pares não classificados no estágio anterior. Como veremos no próximo capítulo, a diferença entre os dois modelos está na utilização ou não dos métodos baseados em alinhamento e no tipo de tarefa como atributo para o treinamento do Classificador Bayesiano Ingênuo (ver seção 5.6).

4.8 Um Modelo Booleano para Pergunta-Resposta

Nesta seção, descrevemos a implementação de um modelo baseado em redes neurais artificiais e transferência de aprendizado para avaliar os pares dos três conjuntos de dados. O modelo implementado foi o RoBERTa[13] (Liu *et al.*, 2019*b*), *A Robustly Optimized BERT Pretraining Approach*, uma otimização do BERT Devlin *et al.* (2018). Para avaliá-lo, utilizamos uma abordagem de Pergunta-Reposta (QA), inspirada em implementações realizadas no corpus BoolQ (Clark *et al.*, 2019*a*). Nas subseções adiante, descrevemos as etapas da implementação do modelo.

4.8.1 Conversão dos Pares em Perguntas Polares

Nesta etapa, convertemos os pares dos três conjuntos de dados em um formato específico para tarefas como a QA. A conversão foi inspirada no BoolQ (Clark *et al.*, 2019*a*), um conjunto de dados para perguntas Sim/Não baseado na tarefa de NLI. O BoolQ é um conjunto de dados para o inglês composto por 16.000 perguntas polares sobre textos da Wikipédia. Na nossa conversão, utilizamos as premissas como textos para as repostas e convertemos as hipóteses em perguntas polares.

[13] A implementação realizada neste trabalho foi adaptada de: https://medium.com/@michelivincent

A conversão foi realizada em duas etapas. Na primeira, utilizamos o Benepar[14], *Berkeley Neural Parser*, para realizar uma análise sintática que nos permitisse reconhecer constituintes das hipóteses. Na segunda, utilizamos a análise de dependências feita com o spaCy para identificar e movimentar os verbos necessários à composição das perguntas. Os exemplos adiante apresentam pares convertidos para o formato de QA com perguntas polares.

(14) ID: "759", Classification: "TRUE" Task: CD
P: Mexico City lies in an endorheic (characterized by interior drainage) basin at an altitude of approximately 7,350 feet (2,240 metres).
H: Mexico City is about 7,350 feet (2,240 meters) above sea level.
Pergunta: Is Mexico City about 7,350 feet (2,240 meters) above sea level?
Resposta: Yes.

(15) ID: "9", Classification: "FALSE" Task: IE
P: Minton's first major part in England was as Maggie Dempster in the premiere of Nicholas Maw's One Man Show. Shortly thereafter, she became a regular member of the company of the Royal Opera House Covent Garden.
H: Maggie Dempster was a member of the company of the Royal Opera House Covent Garden.
Pergunta: Was Maggie Dempster a member of the company of the Royal Opera House Covent Garden?
Resposta: No.

Em (14), temos um exemplo de conversão para um par do RTE-1. Na hipótese em questão, movimentamos o verbo de ligação para o início da pergunta, de modo a criar uma questão polar. Como o par foi classificado como verdadeiro para o acarretamento, a resposta para a pergunta é "Yes". Em (15), por sua vez, temos um par do RTE-3. A resposta para a pergunta polar, dessa vez, é "No", pois trata-se de um par classificado como falso para o acarretamento. O

[14] https://parser.kitaev.io/

pseudocódigo em 4.14 representa o algoritmo implementado na execução dessa tarefa.

O algoritmo 4.14 recebe um par com sua respectiva classificação para acarretamento como argumento. As premissas são tomadas em suas respectivas formas originais como textos base para as respostas das perguntas. As hipóteses são convertidas em perguntas de acordo com o tipo de verbo que possuem. Caso possuam verbos auxiliares ou verbos de ligação, esses verbos são movidos para o início da sentença e um ponto de interrogação é adicionado ao final. Caso contrário, uma partícula auxiliar no tempo passado é adicionada ao início da sentença, o verbo principal é lematizado e um ponto de interrogação também é adicionado ao final. Por fim, as etiquetas de classificação são convertidas em "Yes" ou "No" de acordo com a classificação para acarretamento.

4.8.2 Hiperparâmetros do Modelo

Em nossa implementação, utilizamos *RoBERTa Large*[15], modelo pré-treinado com 355 milhões de parâmetros. A otimização utilizada foi a Adam (Kingma & Ba, 2014) com os seguintes hiperparâmetros: $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e\text{-}6$, Decaimento = 0.01. A taxa de aprendizado foi mantida como recomendada em Clark *et al.* (2019*a*), isto é, 1e-5, o lote, *bacth*, escolhido teve tamanho 10 e o número de épocas também foi 10.

As sentenças dos pares foram truncadas com um tamanho máximo de 256 *tokens*, separadas com os tokens iniciais e finais que o RoBERTa atribui, a saber, <S> e </S>, respectivamente. Os *tokens* foram mapeados para seus respectivos índices. Por fim, utilizamos as máscaras de atenção para discriminar *tokens* relevantes de *tokens* aos quais utilizamos o *padding*. Com isso, conseguimos converter os dados dos corpora em atributos necessários ao RoBERTa. As conversões para tensores foram realizadas com o Pytorch[16].

[15] https://huggingface.co/roberta-large

[16] https://pytorch.org/

4.8.3 Treinamento

Com a finalidade de obter uma base de dados suficiente para o treinamento do modelo, combinamos os três conjuntos de validação oferecidos nos três primeiros anos do *Pascal RTE Challenge*. Conforme descrevemos na Seção 4.1, o RTE-1 possui um conjunto de validação de 576 pares, o RTE-2 e o RTE-3 um de 800 pares cada. Isso nos possibilitou formar um conjunto composto por 2.176 pares. O novo conjunto de treinamento passou pelo mesmo procedimento apresentado em 4.8.1, isto é, as premissas foram utilizadas como textos base para as perguntas, compostas pelas hipóteses, e as respostas foram extraídas das classificações dos pares.

Para o treinamento do modelo, utilizamos o algoritmo de retropropagação, *backpropagation*, e o *gradient clipping*, disponíveis na API do Pytorch. Para fazer as previsões das respostas, utilizamos uma função de ativação sigmoide nos valores para *logits* retornados pelo modelo. A função sigmoide é dada pela seguinte formula:

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{4.5}$$

A função sigmoide nos possibilita mapear um valor real de z em um intervalo entre $[0, 1]$. Como a formula apresenta, a função $\phi(z)$ consiste em dividir 1 por ele mesmo e somá-lo ao número de Euler elevado a $-z$. Ao considerarmos o tipo de resposta que precisamos, isto é, Yes/No, vemos que essa função pode nos oferecer *outputs* similares aos da classificação com o Classificador Bayesiano Ingênuo, isto é, probabilidades. Na Figura 4.1 temos o gráfico para a função.

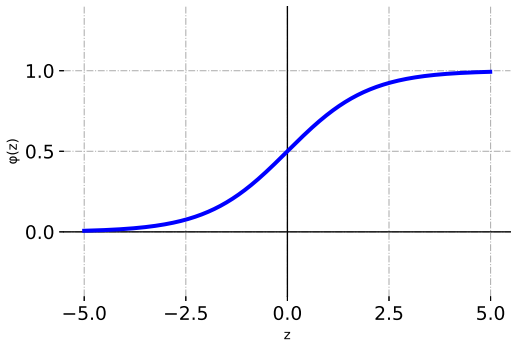


Figura 4.1: Função sigmoide

O gráfico nos apresenta a típica curva em S de onde o nome "sigmoide" é derivado. Ao escolher esse tipo de função de ativação, pretendíamos adequar os *outputs* do modelo às necessidades da tarefa. Há outras funções de ativação, como a softmax, que retornam valores probabilísticos como *outputs*. Para nossos objetivos, modelar *outputs* binários na saída do modelo, a sigmoide se mostrou mais adequada.

4.9 Avaliação dos Modelos

A avaliação dos modelos implementados foi feita em diferentes etapas. Os métodos sem alinhamento, os com alinhamento e o método de representação lógica foram individualmente avaliados de acordo com a acurácia. Essa métrica foi utilizada como critério final de aplicação de cada método. Avaliados por acurácia, os métodos compuseram um algoritmo que foi aplicado aos conjuntos de dados totais. Esse algoritmo, o Classificador Bayesiano Ingênuo e o RoBERTa foram avaliados, então, com as métricas comuns à avaliações em algoritmos de classificação e aprendizado de máquina supervisionado, isto é, a precisão, a cobertura, a acurácia e a medida-F.

A precisão é obtida por meio da proporção de pares classificados como "True" e que, de fato, são classificados como "True" no corpus. A cobertura indica a proporção de pares que são classificados como "True" no corpus e que foram classificados como "True". A acurácia é dada pelo cálculo da proporção de pares

classificados corretamente. A medida-F, por sua vez, é uma média harmônica entre a precisão e a cobertura, sendo obtida pela Equação 4.6:

$$F = 2 \times \frac{\text{Precisão} \times \text{Cobertura}}{\text{Precisão} + \text{Cobertura}} \tag{4.6}$$

No próximo capítulo, apresentaremos os resultados das avaliações dos métodos descritos com base nessas métricas. Apresentaremos, também, uma breve discussão e comparação dos resultados obtidos com os métodos.

Algoritmo 4.12: Coincidência entre Fórmulas

Entrada: Par: $\langle P, H \rangle$
Saída: Par classificado como True ou "Indefinido"

```
para cada ⟨P,H⟩ faça
    fórmulas_da_premissa ← lista com fórmulas de P
    fórmulas_da_hipótese ← lista de fórmulas de H
    total ← 0
    coincidência ← 0
    para cada fórmula da premissa em fórmulas_da_eremissa faça
        para cada item em fórmula_da_premissa faça
            total ← += 1
        fim
    fim
    para cada fórmula da hipótese em fórmulas_da_hipótese faça
        para cada item em fórmula_da_hipótese faça
            se item em fórmula_da_premissa então
                coincidência ← += 1
            senão
                coincidência ← += 0
            fim
        fim
    fim
    limiar ← coincidência/total
    se limiar ≥ 0.60 então
        retorna True
    senão
        retorna "Indefinido"
    fim
fim
```

Algoritmo 4.13: Modelo Final

Entrada: $\langle Conjunto\ de\ Dados \rangle$
Saída: Acurácia, Cobertura, Precisão e Medida-F

```
para cada ⟨Conjunto de Dados⟩ faça
    Pré-processamento
    Regras de Classificação
    Classificador Bayesiano Ingênuo
    Avaliação do Modelo
fim
```

Algoritmo 4.14: Criação de Perguntas Polares

Entrada: Par: $\langle P, H, \text{Classificação} \rangle$
Saída: texto, pergunta_polar, resposta

```
para cada ⟨P, H, Classificação⟩ faça
    para cada P faça
        texto ← P
        para cada H faça
            se H possui verbo auxiliar ou verbo de ligação então
                verbo ← verbo auxiliar ou verbo de ligação
                hipótese ← H sem verbo auxiliar ou verbo de ligação
                pergunta_polar ← verbo + hipótese + "?"
            senão
                auxiliar ← auxiliar de passado
                hipótese ← H com verbo principal na forma lematizada
                pergunta_polar ← auxiliar + hipótese + "?"
            fim
        fim
        se Classificação == "TRUE" então
            resposta ← "Yes"
        senão
            resposta ← "No"
        fim
    fim
    retorna texto, pergunta_polar, resposta
fim
```

5 Resultados e Análise

No capítulo anterior, apresentamos a metodologia utilizada neste trabalho para tentar solucionar os problemas nos três primeiros conjuntos de dados do *Pascal RTE Challenge*, o RTE-1, o RTE-2 e o RTE-3. Como descrito, testamos diferentes métodos para a classificação dos acarretamentos. Os métodos testados foram baseados em regras de BoW sem alinhamento, regras com alinhamento, representação formal para os pares, classificação probabilística com um Classificador Bayesiano Ingênuo e na tarefa de QA com o RoBERTa (Liu *et al.*, 2019*b*). Nas próximas seções, apresentamos os resultados para cada método testado.

5.1 Avaliação dos Métodos Sem Alinhamento

A primeira etapa de avaliação dos algoritmos se deu por meio da checagem da quantidade de pares cobertos por cada um e por suas respectivas acurácias. As acurácias foram computadas com base na quantidade de pares automaticamente selecionados por cada método e não em relação aos conjuntos de dados totais.

Escolhemos a acurácia para avaliação, por ser a principal métrica utilizada nos desafios até o RTE-5. As aplicações foram realizadas de forma independente em cada um dos conjuntos de dados. No primeiro teste, utilizamos as regras de BoW sem algumas das etapas realizadas no pré-processamento, como o uso de máscaras para entidades nomeadas e conversão de dígitos em formas por extenso. Na Tabela 5.1, podemos observar o critério de classificação de cada algoritmo, bem como a quantidade de pares verdadeiros e falsos no RTE-1.

Na Tabela 5.1, as colunas identificadas como "V" e "F" se referem à classificação oferecida nos pares do RTE-1. A coluna "Critério" identifica a classificação

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	3	21	24	0,03	87%
Coincidência Lexical entre o Par	V	25	6	31	0,03	81%
Hipótese Maior que Premissa	F	10	36	46	0,05	78%
Par com Hipótese Vazia	V	11	5	16	0,02	69%
Inclusão de Entidades Nomeadas	F	45	95	140	0,17	68%
Dígitos Apenas na Premissa	V	93	65	158	0,19	59%

Tabela 5.1: Acurácia por Algoritmo no Primeiro Teste – Métodos sem Alinhamento – RTE-1. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

para o atribuída por cada um dos seis algoritmos. A coluna "n" informa a quantidade de pares cobertos por cada algoritmo. Por meio dela, podemos verificar a discrepância entre a quantidade de pares classificados por cada método. A soma total de pare classificados individualmente por cada algoritmo é de 415 pares. A coluna "Rn", por sua vez, informa a frequência relativa dos pares classificados por cada algoritmo no corpus. Já a coluna "Acurácia" apresenta a avaliação de cada algoritmo de acordo com essa métrica. Lembramos, como mencionado anteriormente, que as acurácias foram computadas com base nos pares classificados por cada algoritmo e não em relação aos conjuntos de dados totais. Os algoritmos apresentados em 5.4 estão dispostos por ordem decrescente de acurácia, critério de aplicação utilizado em testes posteriores.

Na Tabela 5.2 apresentamos os resultados da aplicação dos seis algoritmos no conjunto de teste do RTE-2, também composto por 800 pares. A tabela está organizada com as mesmas informações apresentadas para o RTE-1.

Para o RTE-2, a soma total de pares classificados é 365. Um pouco menos do que no RTE-1. Em ambos os conjuntos de dados, os algoritmos para verificar a coincidência lexical e pares com hipótese maiores alcançaram boas acurácias. No RTE-2, porém, as demais métricas não foram tão expressivas. Em relação ao teste feito no RTE-3, podemos acompanhar os resultados na Tabela 5.3.

Os resultados, em termos de acurácia, são piores no RTE-3, embora o total de pares classificados seja maior do que nos outros corpora. O algoritmo para verificar hipóteses maiores, com boa acurácia nos testes anteriores, não classificou

Algoritmo	Critério	V	F	n	Rn	Acurácia
Coincidência Lexical entre o Par	V	11	1	12	0,01	92%
Hipótese Maior que Premissa	F	1	3	4	0,005	75%
Inclusão de Entidades Nomeadas	F	63	101	104	0,13	60%
Dígitos Diferentes na Hipótese	F	5	7	12	0,01	58%
Par com Hipótese Vazia	V	7	5	12	0,01	58%
Dígitos Apenas na Premissa	V	115	106	221	0,27	52%

Tabela 5.2: Acurácia por Algoritmo no Primeiro Teste – Métodos sem Alinhamento - RTE-2. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	1	18	19	0,02	95%
Inclusão de Entidades Nomeadas	F	63	134	197	0,24	68%
Coincidência Lexical entre o Par	V	13	7	20	0,02	65%
Par com Hipótese Vazia	V	14	10	24	0,03	58%
Dígitos Apenas na Premissa	V	131	104	235	0,29	56%
Hipótese Maior que Premissa	F	0	0	0	0	0%

Tabela 5.3: Acurácia por Algoritmo no Primeiro Teste – Métodos sem Alinhamento - RTE-3. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

nenhum par. Isso se deu, provavelmente, pelas características das premissas do RTE-3.

Após os primeiros testes, realizamos uma segunda etapa de classificação com os algoritmos. Dessa vez, porém, utilizamos algumas das etapas realizadas no pré-processamento. As etapas de pré-processamento selecionadas foram o mascaramento de entidades nomeadas e a conversão de dígitos em suas respectivas formas por extenso. Com os novos testes, esperávamos refinar a classificação, melhorando a cobertura e acurácia dos algoritmos. Na Tabela 5.4 apresentamos os resultados no RTE-1.

No total, os algoritmos classificaram 522 pares, 107 a mais do que os 415 do primeiro teste, o que mostra um ganho na cobertura total. As acurácias, no entanto, não foram tão expressivas. Na verdade, todas diminuíram por relação ao primeiro teste no corpus. As melhoras que esperávamos principalmente para

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	3	17	20	0,02	85%
Hipótese Maior que Premissa	F	14	40	54	0,06	74%
Coincidência Lexical entre o Par	V	23	9	32	0,04	71%
Par com Hipótese Vazia	V	13	6	19	0,02	68%
Dígitos Apenas na Premissa	V	94	67	161	0,20	58%
Inclusão de Entidades Nomeadas	F	107	129	236	0,29	55%

Tabela 5.4: Acurácia por Algoritmo no Segundo Teste – Métodos sem Alinhamento - RTE-1. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

dois algoritmos, o que verifica a coincidência lexical e o que verifica a inclusão de entidades nomeadas, não se confirmaram. Em relação ao RTE-2, os resultados do segundo teste podem ser acompanhados na Tabela 5.5.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Coincidência Lexical entre o Par	V	11	0	11	0,01	100%
Par com Hipótese Vazia	V	10	3	13	0,01	77%
Hipótese Maior que Premissa	F	2	5	7	0,008	71%
Dígitos Diferentes na Hipótese	F	5	9	14	0,01	64%
Inclusão de Entidades Nomeadas	F	94	105	199	0,24	53%
Dígitos Apenas na Premissa	V	113	106	219	0,27	52%

Tabela 5.5: Acurácia por Algoritmo no Segundo Teste – Métodos sem Alinhamento - RTE-2. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Os resultados são menos discrepantes em relação ao primeiro teste no RTE-2. A cobertura total, mais uma vez, aumentou. Dessa vez, os algoritmos classificaram 463 pares. Ao contrário do que ocorreu no RTE-1, o refinamento realizado com as etapas de pré-processamento possibilitou melhores resultados. Ao menos para classificação baseada na coincidência lexical, os resultados foram melhores. Já a classificação baseada na inclusão de entidades nomeadas, assim como no RTE-1, piorou por relação ao primeiro teste.

Na Tabela 5.5 também podemos observar o que parece ser uma recorrência na classificação dos algoritmos no segundo teste. Os resultados apresentam algumas

semelhanças com aqueles atingidos no RTE-1. Percebemos, por exemplo, que os quatro primeiros algoritmos, embora em ordem diferente de acurácia, são os mesmos para ambos os conjuntos de dados.

Por fim, apresentamos a Tabela 5.6 com os resultados atingidos no conjunto de teste do RTE-3. O RTE-3 é, dos três conjuntos selecionados nesta pesquisa, aquele que tem características mais diferentes, pois possui premissas maiores do que os outros, traço que poderia influenciar na cobertura de critérios baseados no tamanho da hipótese, na coincidência lexical e, claro, em métodos de alinhamento.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	2	18	20	0,02	90%
Hipótese Maior que Premissa	F	2	7	9	0,01	78%
Par com Hipótese Vazia	V	20	10	30	0,03	67%
Coincidência Lexical entre o Par	V	16	11	27	0,03	59%
Inclusão de Entidades Nomeadas	F	142	170	312	0,39	55%
Dígitos Apenas na Premissa	V	119	102	221	0,27	53%

Tabela 5.6: Acurácia por Algoritmo – Métodos sem Alinhamento - RTE-3. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Um primeiro dado que chama atenção é aquele que diz respeito ao algoritmo para classificar pares com hipóteses maiores, que obteve boa acurácia, embora com baixa cobertura. Em relação ao primeiro teste, as etapas de mascaramento e conversão de dígitos modificaram algumas hipóteses, tornando-as maiores do que as premissas. Consequentemente, houve uma melhora em relação ao primeiro teste no RTE-3.

Também podemos constatar recorrências em relação aos segundos testes realizados no RTE-1 e no RTE-2. Temos, por exemplo, regras como "Dígitos Diferentes na Hipótese" e "Hipótese Maior que Premissa" entre as acurácias mais altas. No caso da última, sua cobertura, embora baixa, ficou próxima da obtida no RTE-2.

As seis tabelas podem ser analisadas ao menos de duas maneiras. Na primeira análise, percebemos que as etapas de pré-processamento adotadas, embora refinem a análise dos algoritmos, tendem a diminuir as acurácias. Nos testes

com as entidades nomeadas, provavelmente, ocorrências de *strings* e *substrings* beneficiaram o algoritmo no primeiro teste. Essas entidades seriam consideradas diferentes na primeira análise. No segundo teste, por outro lado, essa distinção se perdeu, uma vez que *tokens* desse tipo receberiam uma mesma máscara. Nos testes com conversão de dígitos em formas por extenso, dígitos como 1794 foram convertidos em *one thousand*, *seven hundred and ninety-four*. Com isso, os textos ganharam novos *tokens* e podem ter ficado menos coincidentes. Em um cenário como esse, o que seria um único *token*, ou dígito, para o cômputo da coincidência, foi convertido em outros seis. Na segunda análise, podemos verificar os dois testes em sua totalidade. Percebemos, por exemplo, que a acurácia tende a diminuir com o aumento da quantidade de pares classificados por cada algoritmo. Para alguns algoritmos, como aquele que verifica a quantidade de hipóteses maiores e atribui a classificação "False", esperávamos uma baixa cobertura, por razões elencadas anteriormente. Para critérios como a coincidência lexical, a cobertura também pode ter sido limitada pelas características lexicais dos conjuntos de dados, não restritas à co-ocorrência entre *tokens*, e pelas características sintáticas dos pares.

A aplicação dos algoritmos possibilita, ainda, fazer observações sobre a complexidade dos problemas apresentados pelo conjuntos de dados. Os acarretamentos podem estar relacionados à informações que não são dispostas continuamente nos textos dos pares (Androutsopoulos & Malakasiotis, 2010). Consequentemente, ao utilizar regras baseadas em um BoW, perdemos informações sintáticas e semânticas importantes para as avaliações. Apesar das baixas coberturas individuais, os algoritmos obtiveram valores razoáveis para acurácia. Aplicados de acordo com essa métrica, a soma total de pares classificados pelos algoritmos fica acima de 50% nos três conjuntos de dados.

5.2 Avaliação dos Métodos com Alinhamento

Nesta seção, apresentamos a avaliação dos quatro métodos baseados em alinhamento. Assim como nos testes anteriores, os algoritmos foram aplicados individualmente e avaliados por acurácia em cada conjunto de dados. Mais uma

vez, as acurácias foram computadas com base nos pares classificados por cada método. Em 5.7 apresentamos os resultados para o RTE-1.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Co-ocorrência entre Objetos	V	26	15	41	0,05	63%
Coincidência entre Triplas	V	35	23	58	0,07	60%
Co-ocorrência entre Sujeitos	V	72	58	130	0,16	55%
Inclusão de Triplas ou Duplas	V	47	39	86	0,10	55%

Tabela 5.7: Acurácia por Algoritmo – Métodos com Alinhamento - RTE-1. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

No total, isto é, na soma dos valores na coluna "n", os métodos baseados em alinhamento permitiram classificar 315 pares dos 800 presentes no conjunto de teste do RTE-1. As acurácias, no entanto, ficaram pouco acima dos 50% em dois dos testes. Os resultados são próximos dos obtidos no RTE-2, apresentados em 5.8.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Co-ocorrência entre Objetos	V	8	3	11	0,01	73%
Co-ocorrência entre Sujeitos	V	62	36	98	0,12	63%
Inclusão de Triplas e Duplas	V	41	28	69	0,08	59%
Coincidência entre Triplas	V	19	14	33	0,04	58%

Tabela 5.8: Acurácia por Algoritmo – Métodos com Alinhamento - RTE-2. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Como podemos perceber, em ambos os testes, o algoritmo que verifica objetos iguais obteve a melhor acurácia. Os resultados no RTE-2 também são relativamente melhores do que os no RTE-1, embora a soma total de pares classificados pelos algoritmos seja menor, a saber, 211 pares. Esses resultados apresentam maior discrepância quando observamos, na Tabela 5.9, a avaliação dos métodos no RTE-3.

No RTE-3, por sua vez, os métodos possibilitaram classificar 196 pares (soma total da coluna "n"). O critério para classificar pares com objetos iguais, com

Algoritmo	Critério	V	F	n	Rn	Acurácia
Coincidência entre Triplas	V	18	9	27	0,03	67%
Co-ocorrência entre Sujeitos	V	69	45	114	0,14	61%
Inclusão de Triplas ou Duplas	V	26	19	45	0,05	58%
Co-ocorrência entre Objetos	V	5	5	10	0,01	50%

Tabela 5.9: Acurácia por Algoritmo - Métodos com Alinhamento - RTE-3. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

a melhor acurácia nos testes anteriores, não foi efetivo no RTE-3 e, além disso, teve baixa cobertura.

5.3 Avaliação do Método Baseado em Representação Lógica

Apesar de a variabilidade linguística dos RTEs ser um fator limitador para modelos baseados nesse tipo de representação, o DepCCG e o CCG2Lambda possibilitaram converter um bom número de pares nos três conjuntos de dados. No total, 783 pares do RTE-1 foram convertidos, do RTE-2 foram 788 e do RTE-3 foram 754. A quantidade de pares é significativa, sobretudo quando consideramos a variabilidade linguística dos RTEs e os obstáculos que ela oferece à formalizações lógicas (Bos, 2014). Na Tabela 5.10 temos as acurácias do método para os três corpora. Assim como nos métodos anteriores, computamos as acurácias com base nos pares classificados pelo algoritmo.

Corpus	Cn	Critério	V	F	n	Rn	Acurácia
RTE-1	783	V	35	18	53	0,06	66%
RTE-2	788	V	8	5	13	0,01	61%
RTE-3	754	V	4	6	10	0,01	40%

Tabela 5.10: Acurácia do Método de Representação Lógica. Cn = quantidade de pares convertidos para forma lógica pelo CCG2Lambda, n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério nos pares convertidos para forma lógica (frequência relativa de n).

Como podemos observar, os resultados são melhores em termos de cobertura e acurácia para o RTE-1. O método foi testado com base em um limiar de coincidência com valor de 0.6, conforme descrevemos no Cap. 4. Avaliamos, também, o algoritmo com limiares diferentes. No entanto, limiares maiores permitiam obter pouco menos de 10 pares para cada corpus, embora a acurácia do algoritmo aumentasse.

5.4 Algoritmo Regras de Classificação

A partir das avaliações anteriores, decidimos compor um algoritmo com cada um dos métodos. Denominamos esse algoritmo Regras de Classificação. Para avaliá-lo, realizamos três testes. No primeiro, avaliamos o algoritmo somente com os métodos sem alinhamento e sem as etapas de mascaramento de entidades nomeadas e conversão de dígitos em formas por extenso. No segundo, avaliamos os algoritmos sem alinhamento com essas etapas de pré-processamento. No terceiro, utilizamos as regras do BoW com as etapas de refinamento do pré-processamento e também os métodos com alinhamento para a avaliação. Nosso objetivo era comparar as aplicações para verificar quais seriam os melhores resultados. Ao contrário da avaliação individual de cada método (acurácia), o algoritmo Regras de Classificação foi avaliado de acordo com a cobertura, a acurácia, a precisão e a medida-F. Em razão de sua baixa acurácia e cobertura no RTE-3, optamos por não incluir o método de representação lógica nesse algoritmo.

Na Seção 5.1, vimos que os métodos sem alinhamento e etapas de refinamento no pré-processamento classificaram 415 no RTE-1, 365 no RTE-2 e 495 no RTE-3. Esses valores, como mencionamos anteriormente, indicam a soma total dos pares classificados por cada algoritmo. Esperávamos que, provavelmente, algoritmos diferentes classificariam pares iguais. Em razão disso, utilizamos a acurácia como critério de ordem para a aplicação. O algoritmo Regras de Classificação, portanto, inicia a classificação dos pares aplicando cada um dos algoritmos apresentados nas seções anteriores por ordem de acurácia. Ao fim da aplicação, o algoritmo Regras de Classificação é avaliado com base na precisão, acurácia,

cobertura e medida-F. Na Tabela 5.11, observamos os resultados obtidos no RTE-1 para o primeiro teste.

Corpus	n	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	342	70%	72%	67%	69%
RTE-2	322	57%	71%	55%	62%
RTE-3	408	65%	68%	61%	64%

Tabela 5.11: Avaliação do Algoritmo Regras de Classificação com Métodos sem Alinhamento no Primeito Teste. n = número de pares no conjunto de dados no qual o algoritmo foi avaliado.

A Tabela 5.11 mostra que dos 415 pares classificados, 73 eram cobertos por ao menos dois algoritmos. No RTE-2 e no RTE-3 o cenário é semelhante, neste 87 pares eram classificados por ao menos dois algoritmos e naquele eram 43 pares. Os resultados são expressivos principalmente no RTE-1 e no RTE-3, nos quais o algoritmo Regras de Classificação atingiu mais de 60% para cada métrica.

No segundo teste, aplicamos os algoritmos com as etapas de refinamento do pré-processamento. Conforme apresentamos na Seção 5.1, os métodos sem alinhamento classificaram um total de 522 pares no RTE-1, 463 no RTE-2 e 619 no RTE-3. Na Tabela 5.12 podemos observar a performance do algoritmo em cada conjunto de dados no teste.

Corpus	n	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	415	59%	71%	58%	64%
RTE-2	399	57%	74%	57%	65%
RTE-3	474	59%	66%	57%	61%

Tabela 5.12: Avaliação do Algoritmo Regras de Classificação com Métodos sem Alinhamento no Segundo Teste. n = número de pares no conjunto de dados no qual o algoritmo foi avaliado.

Mais uma vez, alguns algoritmos classificavam pares iguais. Dessa vez, 107 pares eram classificados por ao menos dois algoritmos no RTE-1, 64 no RTE-2 e 145 no RTE-3. Os resultados, no entanto, diminuíram consideravelmente no RTE-1 e no RTE-3 em relação ao teste anterior.

No terceiro teste, por fim, combinamos as regras do BoW com mascaramento de entidades nomeadas, conversão de dígitos em formas por extenso e com as

regras com alinhamento. Como algumas delas obtiveram acurácias melhores do que os métodos baseados em BoW, reorganizamos a ordem de aplicação do algoritmo Regras de Classificação. Em 5.13 podemos observar a nova ordem de aplicação para o RTE-1.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	3	17	20	0,02	85%
Hipótese Maior que Premissa	F	14	40	54	0,06	74%
Coincidência Lexical entre o Par	V	23	9	32	0,04	71%
Par com Hipótese Vazia	V	13	6	19	0,02	68%
Co-ocorrência entre Objetos	V	26	15	41	0,05	63%
Coincidência entre Triplas	V	35	23	58	0,07	60%
Dígitos Apenas na Premissa	V	94	67	161	0,20	58%
Co-ocorrência entre Sujeitos	V	72	58	130	0,16	55%
Inclusão de Triplas ou Duplas	V	47	39	86	0,10	55%
Inclusão de Entidades Nomeadas	F	107	129	236	0,29	55%

Tabela 5.13: Acurácia por Algoritmo – Métodos com e sem Alinhamento - RTE-1. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

A Tabela 5.13 apresenta os métodos organizados por ordem de acurácia e, portanto, de aplicação. A soma total de pares classificados individualmente é 837, o que já nos indica que boa parte desses pares é classificada por um mesmo método. Já no RTE-2, podemos observar, na Tabela 5.14, a nova organização com a combinação dos métodos.

No total, a soma das aplicações é de 674 no RTE-2, pouco menos do que no RTE-1. Ainda assim, deduzimos que parte desses pares foi classificada por mais de um método. No RTE-3, por sua vez, a organização por ordem de acurácia é apresentada na Tabela 5.15.

A Tabela 5.15 mostra que, de uma maneira geral, as menores acurácias foram obtidas no RTE-3. Por fim, uma vez que os métodos foram organizados por ordem de acurácia, realizamos o terceiro teste, aplicando o algoritmo Regras de Classificação nos conjuntos de dados e avaliando-o. Na Tabela 5.16, apresentamos seus resultados.

Algoritmo	Critério	V	F	n	Rn	Acurácia
Coincidência Lexical entre o Par	V	11	0	11	0,01	100%
Par com Hipótese Vazia	V	10	3	13	0,01	77%
Co-ocorrência de Objetos	V	8	3	11	0,01	73%
Hipótese Maior que Premissa	F	2	5	7	0,008	71%
Dígitos Diferentes na Hipótese	F	5	9	14	0,01	64%
Co-ocorrência de Sujeitos	V	62	36	98	0,12	63%
Inclusão de Triplas ou Duplas	V	41	28	69	0,08	59%
Coincidência entre Triplas	V	19	14	33	0,04	58%
Inclusão de Entidades Nomeadas	F	94	105	199	0,24	53%
Dígitos Apenas na Premissa	V	113	106	219	0,27	52%

Tabela 5.14: Acurácia por Algoritmo – Métodos com e sem Alinhamento - RTE-2. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Ao analisarmos as métricas obtidas, percebemos que não houve alteração significativa em algumas delas por relação aos resultados apresentados na Tabela 5.12. Os resultados são melhores, principalmente para o RTE-3, para o qual a acurácia, a precisão e a medida-F ficaram maiores do que no primeiro teste. No entanto, se compararmos os resultados obtidos no terceiro teste com aqueles obtidos no primeiro, constatamos que a performance do algoritmo diminuiu. Com isso, concluímos que refinar a análise dos algoritmos por meio do pré-processamento teve um efeito negativo em termos de resultados.

5.5 Avaliação do Classificador Bayesiano Ingênuo

Retomando o que apresentamos no capítulo anterior, para cada teste realizado com o Classificador Bayesiano Ingênuo, separamos aleatoriamente os pares de cada conjunto de dados em 80% para treino e 20% para teste. Utilizamos o classificador para três testes. No primeiro, testamos o classificador nos conjuntos de dados totais. No segundo, o teste foi feito nos pares não classificados pelo algoritmo Regras de Classificação sem etapas de refinamento com o pré-processamento e sem regras de alinhamento. Nesse teste, também utilizamos as tarefas como atributos para o treino do classificador. No terceiro, testamos o classificador nos pares não classificados pelo algoritmo Regras de Classificação,

Algoritmo	Critério	V	F	n	Rn	Acurácia
Dígitos Diferentes na Hipótese	F	2	18	20	0,02	90%
Hipótese Maior que Premissa	F	2	7	9	0,01	78%
Coincidência entre Triplas	V	18	9	27	0,03	67%
Par com Hipótese Vazia	V	20	10	30	0,03	67%
Co-ocorrência entre Sujeitos	V	69	45	114	0,14	61%
Coincidência Lexical entre o Par	V	16	11	27	0,03	59%
Inclusão de Triplas ou Duplas	V	26	19	45	0,05	58%
Inclusão de Entidades Nomeadas	F	142	170	312	0,39	55%
Dígitos Apenas na Premissa	V	119	102	221	0,27	53%
Co-ocorrência entre Objetos	V	5	5	10	0,01	50%

Tabela 5.15: Acurácia por Algoritmo – Métodos com e sem Alinhamento - RTE-3. n = número de pares nos conjuntos de dados que satisfazem aos critérios estabelecidos, Rn = representatividade do critério no corpus (frequência relativa de n).

Corpus	n	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	506	59%	71%	58%	64%
RTE-2	458	57%	74%	57%	65%
RTE-3	538	60%	73%	57%	64%

Tabela 5.16: Avaliação do Algoritmo Regras de Classificação com e sem Métodos de Alinhamento no Terceiro Teste. n = número de pares no conjunto de dados no qual o algoritmo foi avaliado.

dessa vez composto pelo BoW, pela etapa de refinamento e pelas regras de alinhamento. Nesse terceiro teste, não utilizamos os tipos de tarefa como atributo para o treinamento. Na Tabela 5.17 apresentamos os resultados para a avaliação nos conjuntos totais de teste.

Corpus	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	49%	67%	48%	56%
RTE-2	52%	39%	50%	44%
RTE-3	58%	65%	57%	61%

Tabela 5.17: Avaliação do Classificador Bayesiano Ingênuo no RTE-1, RTE-2 e RTE-3 – Teste 1.

O Classificador Bayesiano Ingênuo não foi efetivo na classificação total dos conjuntos de dados quando treinado com os atributos que selecionamos. O desempenho do classificador apresenta uma melhora quando avaliado nos pares não classificados pelo algoritmo Regras de Classificação sem regras de alinhamento, máscaras de entidades nomeadas e conversão de dígitos em formas por extenso, como podemos observar na Tabela 5.18.

Corpus	n	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	458	61%	55%	63%	59%
RTE-2	478	58%	54%	54%	54%
RTE-3	392	61%	64%	62%	63%

Tabela 5.18: Avaliação do Classificador Bayesiano Ingênuo nos pares não classificados pelo Algoritmo Regras de Classificação – Teste 2. n = número de pares no conjunto de dados no qual o algoritmo foi avaliado.

Os melhores resultados, como apresentado na Tabela 5.18, foram alcançados no RTE-3 para o segundo teste. Por fim, apresentamos os resultados no terceiro teste por meio da Tabela 5.19.

Corpus	n	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	294	50%	57%	59%	58%
RTE-2	342	49%	89%	44%	59%
RTE-3	262	60%	68%	60%	64%

Tabela 5.19: Avaliação do Classificador Bayesiano Ingênuo nos pares não classificados pelo Algoritmo Regras de Classificação – Teste 3. n = número de pares no conjunto de dados no qual o algoritmo foi avaliado.

Por meio da Tabela 5.19, observamos que a acurácia ficou em 50% no RTE-1 e abaixo dessa porcentagem no RTE-2. Apenas no RTE-3 a acurácia ficou próxima dos valores alcançados pelo algoritmo Regras de Classificação. Em relação ao segundo teste, o classificador não obteve uma performance tão díspar. Os melhores resultados continuaram no RTE-3. Nos outros dois corpora, as diferenças não foram perceptíveis.

A partir dos testes, decidimos, como veremos na próxima seção, utilizar o classificador como parte de um modelo final. Sendo assim, procuramos verificar

quais atributos tiveram maior influência no desempenho do classificador. Como anteriormente descrito, a lista de atributos foi composta pela comparação entre a quantidade de palavras relevantes nos dois textos do par (premissa e hipótese), pelo grau de coincidência entre os pares, que poderia ser baixo ou não e, no caso do segundo teste, pelo tipo de tarefa. A Tabela 5.20 resume a informatividade dos atributos no segundo teste para o RTE-1.

Atributo	% T	% F	Proporção
MT	0,052	0,023	2,22
CD	0,057	0,029	1,94
Hipótese Com Mais Palavras Relevantes	0.013	0.025	1,90
PP	0.020	0,031	1,56
IE	0.047	0,069	1,46
IR	0.042	0,049	1,17
RC	0.072	0,084	1,16
Coincidência Não-Baixa	0,143	0,165	1,15
Coincidência Baixa	0,189	0,167	1,13
QA	0,040	0,045	1,12
Premissa Com mais Palavras Relevantes	0,269	0,254	1,06
Mesma Quantidade de Palavras Relevantes	0,0508	0,0533	1,05

Tabela 5.20: Atributos Mais Informativos no Teste 2 – RTE-1.

O atributo mais informativo para o classificador é um tipo de tarefa. Ao filtrarmos os pares classificados pelo algoritmo Regras de Classificação, os pares restantes não ficaram balanceados em relação aos tipos de tarefa. Esse atributo pode ter melhorado o desempenho do Classificador Bayesiano Ingênuo. Já para o RTE-2, podemos verificar a informatividade dos atributos por meio da Tabela 5.21.

No RTE-2, o atributo mais informativo no segundo teste não foi um tipo de tarefa. Ainda assim, as tarefas estiveram entre os três atributos que mais influenciaram o classificador. Por fim, para o segundo teste, apresentamos os atributos mais informativos por meio da Tabela 5.22.

Assim, como no RTE-1, o traço mais informativo no segundo teste para o RTE-3 foi um tipo de tarefa, a saber, QA. No RTE-3, há também um dado novo por relação às informações apresentadas anteriormente: a informatividade do atributo "Hipótese Com Mais Palavras Relevantes". Nos dois testes anteriores,

Atributo	% T	% F	Proporção
Hipótese Com Mais Palavras Relevantes	0,006	0,019	2,90
IE	0,079	0,105	1,32
SUM	0,088	0,068	1,30
Mesma Quantidade de Palavras Relevantes	0,034	0,026	1,26
Coincidência Baixa	0,122	0,145	1,19
Coincidência Não-Baixa	0,210	0,188	1,12
QA	0,076	0,073	1,04
IR	0,088	0,086	1,03
Premissa Com mais Palavras Relevantes	0,292	0,286	1,02

Tabela 5.21: Atributos Mais Informativos no Teste 2 – RTE-2.

Atributo	% T	% F	Proporção
QA	0,082	0,014	5,57
Hipótese Com Mais Palavras Relevantes	0,015	0,007	2,03
IR	0,056	0,111	1,98
Coincidência Não-Baixa	0,166	0,130	1,27
Coincidência Baixa	0,166	0,202	1,21
SUM	0,106	0,125	1,18
Mesma Quantidade de Palavras Relevantes	0,033	0,039	1,17
IE	0,088	0,081	1,08
Premissa Com mais Palavras Relevantes	0,284	0,286	1,01

Tabela 5.22: Atributos Mais Informativos no Segundo Teste – RTE-3.

esse atributo influenciou o classificador a atribuir a classificação "False". No RTE-3, ao contrário, o atributo foi informativo para a classificação "True".

Em relação ao terceiro teste, isto é, aquele nos pares não classificados pelo algoritmo Regras de Classificação com regras de BoW, alinhamento e etapas de pré-processamento para refinar a análise, apresentamos os atributos mais informativos na Tabela 5.23.

Atributo	% T	% F	Proporção
Hipótese Com Mais Palavras Relevantes	0,023	0,036	1,54
Mesma Quantidade de Palavras Relevantes	0,051	0,076	1,48
Coincidência Não-Baixa	0,186	0,208	1,11
Premissa Com Mais Palavras Relevantes	0,425	0,388	1,10
Coincidência Baixa	0,313	0,292	1,07

Tabela 5.23: Atributos Mais Informativos no Terceiro Teste – RTE-1.

O atributo mais informativo para a classificação "False" é o que diz respeito a mais palavras relevantes na hipótese, assim como para o RTE-1 e o RTE-2 nos testes anteriores. É interessante notar que hipóteses com essas características possuem mais palavras do que as premissas. Essas palavras, sendo relevantes de acordo com os valores de IDF, parecem diminuir as chances de acarretamento por trazerem informações novas por relação às premissas.

Na Tabela 5.24, apresentamos os atributos mais informativos para a classificação dos pares restantes do RTE-2 no terceiro teste. Como podemos notar, há uma tendência em relação a alguns traços mais informativos para o classificador.

Atributo	% T	% F	Proporção
Hipótese Com Mais Palavras Relevantes	0,007	0,041	5,20
Premissa Com mais Palavras Relevantes	0,452	0,417	1,08
Mesma Quantidade de Palavras Relevantes	0,039	0,041	1,04
Coincidência Baixa	0,318	0,318	1,00
Coincidência Não-Baixa	0,181	0,181	1,00

Tabela 5.24: Atributos Mais Informativos – RTE-2.

Assim como no RTE-1, o atributo mais informativo no RTE-2 é "Hipótese Com Mais Palavras Relevantes", influenciado o classificador a atribuir a classificação "False". A tabela também mostra a discrepância entre a informatividade desse traço e os demais. Os outros quatro atributos praticamente não influenciam o classificador.

Por fim, na Tabela 5.25, apresentamos os atributos mais informativos para o Classificador Bayesiano Ingênuo nos pares restantes do RTE-3 para o terceiro

teste. Os valores apresentados confirmam a tendência do atributo mais informativo presente nos testes dos outros dois conjuntos de dados.

Atributo	% T	% F	Proporção
Hipótese Com Mais Palavras Relevantes	0,012	0,032	2,50
Coincidência Baixa	0,112	0,204	1,82
Mesma Quantidade de Palavras Relevantes	0,431	0,591	1,37
Coincidência Não-Baixa	0,387	0,295	1,31
Premissa Com mais Palavras Relevantes	0,444	0,408	1,09

Tabela 5.25: Atributos Mais Informativos no Terceiro Teste – RTE-3.

Como podemos notar, os atributos são mais balanceados no RTE-3 nesse terceiro teste. Mais uma vez, "Hipótese Com Mais Palavras Relevantes" é o atributo mais informativo para a classificação "False". A tabela também mostra a predominância de atributos que influenciam classificação "False" para os pares. Na próxima seção, apresentamos a avaliação do modelo composto pelo algoritmo Regras de Classificação e pelo Classificador Bayesiano Ingênuo.

5.6 Avaliação das Regras de Classificação e do Classificador Bayesiano Ingênuo

Nesta seção, apresentamos a avaliação final de dois dos modelos implementados por nós, cujo pseudocódigo foi descrito na Seção 4.7. O primeiro modelo é composto pela junção dos métodos de BoW, sem alinhamento e refinamentos por meio do pré-processamento, com o Classificador Bayesiano Ingênuo. Para o Classificador Bayesiano Ingênuo, utilizamos também o tipo de tarefa como atributo de treinamento, tal como no segundo teste apresentado na Seção 5.5 (ver Tabela 5.18). Nomeamos esse modelo como BCBI (Bag-of-Words e Classificador Bayesiano Ingênuo). O segundo modelo é chamado de BACBI (Bag-of-Word, Alinhamento e Classificador Bayesiano Ingênuo). Esse modelo é composto pelas seis regras de BoW com as etapas de refinamento do pré-processamento, com os métodos de alinhamento e com o Classificador Bayesiano Ingênuo. No segundo

modelo, não utilizamos o tipo de tarefa como atributo para o treinamento do classificador (ver Tabela 5.19).

Conforme apresentamos na Seção 4.7, os modelos consistem na aplicação, após a etapa de pré-processamento, do algoritmo Regras de Classificação e do Classificador Bayesiano Ingênuo. O Classificador Bayesiano Ingênuo foi, portanto, utilizado nos pares não classificados no estágio anterior do modelo.

A avaliação consistiu na combinação entre as regras de coincidência do BoW com o Classificador Bayesiano Ingênuo, no caso do BCBI, e na combinação entre as regras do BoW com alinhamento e o Classificador Bayesiano Ingênuo, no caso do BACBI. Estimamos os valores para as métricas com o cálculo da média aritmética dos resultados obtidos nas duas etapas de classificação. A Tabela 5.26 apresenta os resultados do modelo BCBI. Já para o modelo BACBI, os resultados são apresentados na Tabela 5.27.

Corpus	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	65%	64%	65%	64%
RTE-2	57%	63%	55%	58%
RTE-3	63%	66%	62%	64%

Tabela 5.26: Avaliação do Modelo BCBI.

Corpus	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	55%	64%	58%	61%
RTE-2	57%	82%	53%	64%
RTE-3	60%	60%	67%	63%

Tabela 5.27: Avaliação do Modelo BACBI.

Como podemos perceber, o modelo BACBI apresenta resultados razoavelmente bons para métricas como a precisão e a medida-F. A acurácia, no entanto, ficou próxima do acerto aleatório, com pouco mais de 50% no RTE-1 e no RTE-2. Ao compararmos os modelos entre si, verificamos que o BCBI teve uma performance geral melhor do que a do BACBI.

5.7 Avaliação do RoBERTa

Nesta seção apresentamos a avaliação do teste realizado com o RoBERTa (Liu *et al.*, 2019*b*). Conforme descrito na metodologia, realizamos um teste baseado na tarefa de QA, utilizando premissas como textos de base para respostas e convertendo hipóteses em perguntas polares, isto é, perguntas do tipo Sim/Não. A preparação e a implementação do modelo foram inspiradas no BoolQ (Clark *et al.*, 2019*a*), conjunto de dados de pergunta polares para o inglês. Para realizar o teste, treinamos o modelo na base composta pelos conjuntos de validação dos três corpora, o que nos deu um total de 2.176 pares. A Tabela 5.28 apresenta a avaliação do modelo nos conjuntos de dados totais.

Corpus	Acurácia	Precisão	Cobertura	Medida-F
RTE-1	74%	71%	75%	73%
RTE-2	78%	87%	74%	80%
RTE-3	71%	83%	67%	75%

Tabela 5.28: Avaliação do RoBERTa no RTE-1, RTE-2 e RTE-3.

O desempenho do RoBERTa supera todos os anteriores, em todas as medidas. Os valores de acurácia, por exemplo, com exceção do RTE-3, ficaram acima dos melhores obtidos no RTE-1 e no RTE-2. Mesmo sem etapas de refinamento dos parâmetros, o modelo possibilitou alcançar resultados expressivos para os três conjuntos de dados.

5.8 Avaliação nos Desafios

Por fim, apresentamos os resultados comparando-os com aqueles obtidos pelos participantes dos três desafios e por outros trabalhos citados ao longo desta dissertação. A comparação nos permite avaliar os resultados do ponto de vista de seu alcance, isto é, como proposta de solução dos problemas. A Tabela 5.29 apresenta as acurácias dos modelos comparados no RTE-1.

No RTE-1, o modelo BCBI obteve uma acurácia acima do melhor trabalho avaliado, que ficou em 60%, com Delmonte *et al.* (2005). O BACBI, com 55%

de acurácia não teve um desempenho expressivo. O RoBERTa, por sua vez, superou em quase 10% a acurácia do modelo BCBI, evidenciando a efetividade de modelos baseados em transferência de aprendizado nesse tipo de tarefa.

No RTE-2, o primeiro e o segundo resultados, obtidos por Hickl *et al.* (2006) e Tatu *et al.* (2006), respectivamente 75% e 74% de acurácia, já representam avanços significativos por relação ao ano anterior. Na Tabela 5.30, apresentamos a avaliação dos modelos implementados com os trabalhos avaliados no RTE-2.

Na Tabela 5.30, percebemos a discrepância entre os resultados obtidos pelos modelos BCBI e BACBI e as primeiras colocações no RTE-2. De uma maneira geral, boa parte dos resultados nesse conjunto de dados ficou na média obtida no primeiro ano do desafio. Os resultados de Hickl *et al.* (2006) e Tatu *et al.* (2006) destoam dos demais apresentados. O RoBERTa, por sua vez, atingiu resultados superiores às melhores acurácias.

Por fim, apresentamos a Tabela 5.31, composta pelos participantes do RTE-3 e por outros trabalhos citados ao longo desta dissertação. Embora ambos os modelos, BCBI e BACBI, tenham atingido acurácias melhores por relação ao desempenho no RTE-2, mantiveram-se distantes dos melhores resultados no desafio. Mesmo o RoBERTa, que ficou acima dos melhores resultados no RTE-1 e no RTE-2, não superou Hickl & Bensley (2007*a*) e Tatu & Moldovan (2007).

Modelo	**Acurácia**
RoBERTa QA	0,74
BCBI	0,65
Bos & Markert (2005*b*)	0,61
Delmonte *et al.* (2005)	0,60
Glickman *et al.* (2005)	0,59
Bayer *et al.* (2005)	0,59
Raina *et al.* (2005)	0,56
de Salvo Braz *et al.* (2005)	0,56
Herrera *et al.* (2005)	0,56
Newman *et al.* (2005)	0,56
Bentivogli *et al.* (2009)	0,55
Jijkoun *et al.* (2005)	0,55
Kouylekov & Magnini (2005)	0,55
Bos & Markert (2005*a*)	0,55
BACBI	0,55
Bentivogli *et al.* (2009)	0,55
Fowler *et al.* (2005)	0,55
MacCartney (2009) - BoW	0,53
Andreevskaia *et al.* (2005)	0,52
Pazienza *et al.* (2005)	0,52
Akhmatova (2005)	0,51
Wu (2005)	0,51
Pérez & Alfonseca (2005)	0,49

Tabela 5.29: Acurácia dos Modelos Avaliados no RTE-1.

Modelo	**Acurácia**
RoBERTa QA	0,78
Hickl *et al.* (2006)	0,75
Tatu *et al.* (2006)	0,74
Zanzotto *et al.* (2006)	0,64
Adams (2006)	0,62
Bos & Markert (2006)	0,61
Marsi *et al.* (2006)	0,60
Vanderwende *et al.* (2006)	0,60
De Marneffe *et al.* (2006)	0,60
Kouylekov & Magnini (2006)	0,60
Herrera *et al.* (2006)	0,59
Nielsen *et al.* (2006)	0,59
Katrenko *et al.* (2006)	0,59
Burchardt & Frank (2006)	0,59
Inkpen *et al.* (2006)	0,58
Rus (2006)	0,58
Litkowski (2006)	0,58
BCBI	0,57
BACBI	0,57
MacCartney (2009) - BoW	0,57
Ferrández *et al.* (2006)	0,55
Kozareva & Montoyo (2006)	0,55
Schilder & McInnes (2006)	0,55
Delmonte *et al.* (2006)	0,54
Clarke (2006)	0,54
Bentivogli *et al.* (2009)	0,54
Newman *et al.* (2006)	0,54
Nicholson *et al.* (2006)	0,52

Tabela 5.30: Acurácia dos Modelos Avaliados no RTE-2.

Modelo	**Acurácia**
Hickl & Bensley (2007*a*)	0,80
Tatu & Moldovan (2007)	0,72
RoBERTa QA	0,71
Iftene & Balahur-Dobrescu (2007)	0,69
Adams *et al.* (2007)	0,67
Wang & Neumann (2007)	0,66
Zanzotto *et al.* (2007)	0,66
Ferrández *et al.* (2007)	0,65
Blake (2007)	0,65
Li *et al.* (2007)	0,64
BCBI	0,63
Rodrigo *et al.* (2007)	0,63
MacCartney (2009) - BoW	0,63
Burchardt *et al.* (2007)	0,63
Chambers *et al.* (2007)	0,62
Settembre (2007)	0,62
Bentivogli *et al.* (2009)	0,62
Roth & Sammons (2007)	0,62
Malakasiotis & Androutsopoulos (2007)	0,61
Ferrés & Rodríguez (2007)	0,61
Bar-Haim *et al.* (2007)	0,61
Montejo-Ráez *et al.* (2007)	0,60
BACBI	0,60
MacCartney (2009) - Natlog	0,59
Marsi *et al.* (2007)	0,59
Delmonte *et al.* (2007)	0,58
Harmeling (2007)	0,57
Burek *et al.* (2007)	0,55
Bobrow *et al.* (2007)	0,51
Clark *et al.* (2007)	0,50

Tabela 5.31: Acurácia dos Modelos Avaliados no RTE-3.

6

Conclusão

6.1 Discussão sobre a Proposta do Trabalho

Iniciamos este trabalho com a proposta de discutir e comparar diferentes métodos de solução para os problemas de NLI oferecidos pelos três primeiros conjuntos de dados do *Pascal RTE Challenge*, o RTE-1, o RTE-2 e o RTE-3. Para atender a essa proposta, testamos diferentes possibilidades de solução com métodos mais e menos tradicionais nos conjuntos de dados.

Testar métodos baseados na coincidência lexical entre os pares para classificar acarretamentos foi uma estratégia amplamente utilizada pelos participantes do *Pascal RTE Challenge*. Seja como metodologia central ou como método para extração de atributos para classificadores, o grau de coincidência foi quase sempre um caminho explorado. Em verdade, sobretudo no primeiro ano do desafio, metodologias baseadas nesse tipo de análise obtiveram maior êxito na classificação (Dagan *et al.*, 2005).

Utilizar regras de BoW é optar por uma proposta de solução que desconsidera aspectos fundamentais da linguagem humana. Mesmo se pensarmos nas motivações mais comuns para o uso desses métodos, isto é, seu baixo custo computacional e praticidade, deveremos atentar para suas limitações enquanto proposta de solução. Métodos de BoW simples não são suficientes para atingir resultados expressivos na NLI e, particularmente, nos conjuntos de dados do *Pascal RTE Challenge*.

Nossas regras de BoW, por exemplo, não possuem caráter linguístico. Ao considerarmos o teste baseado na diferença entre dígitos, regra com melhor acurácia no RTE-1 e no RTE-3, percebemos que, do ponto de vista linguístico, não há necessariamente uma relação entre dígitos diferentes em um par de premissa-hipótese e o acarretamento. Uma sentença do tipo "João comprou mais de 5 livros" acarreta uma hipótese como "João comprou mais de 3 livros". Podemos dizer o mesmo da regra para classificar pares falsos com base no tamanho da hipótese. Uma hipótese maior que uma premissa pode, de fato, adicionar informações que eliminam um acarretamento. No entanto, sem analisar que tipo de informação é adicionada, não podemos afirmar que esse acréscimo mantém relação com a ausência de acarretamento. Um par como "Pedro perdeu a eleição para João" e "Pedro, que concorreu à eleição, a perdeu para João" possui uma relação de acarretamento, mesmo com uma hipótese maior. Outros contraexemplos poderiam ser encontrados para os demais métodos.

Dos nossos métodos, aqueles que poderiam avaliar os acarretamentos com base em informações não consideradas pelo BoW eram os baseados em alinhamento e representação lógica. A co-ocorrência de sujeitos e objetos, os métodos com as triplas e a representação lógica trouxeram a possibilidade de recuperar informações sintáticas e semânticas importantes para as avaliações. Mesmo assim, a cobertura e acurácia de cada método ficou próxima das atingidas pelo BoW.

Os métodos baseados na extração e comparação das triplas, também fundamentados em coincidência, foram limitados pelas características lexicais dos textos. Por meio do OpenIE, conseguimos recuperar informações importantes sobre aspectos sintáticos e semânticos dos pares. Por vezes, em pares com textos muito discrepantes em relação ao tamanho, obtivemos triplas específicas de partes dos textos com informações relevantes para avaliar os acarretamentos. A comparação final, feita por meio da análise de coincidência, esbarrava na variabilidade lexical dos pares, no entanto.

Em relação às triplas e duplas expandidas, método que poderia minimizar o impacto da variação lexical, a limitação pode ter sido ocasionada por pelo menos dois motivos. O primeiro diz respeito ao modo como expandimos essas estruturas. Restringimos a expansão apenas à oração principal das premissas e,

procedendo assim, não cobrimos pares em que o acarretamento poderia estar relacionado à informações fora da oração principal. O segundo diz respeito à variabilidade sintática dos pares, o que dificulta seu parseamento automático. A variabilidade sintática pode, inclusive, ter sido um fator limitador para os testes de co-ocorrências. Enquanto as hipóteses são, em sua maioria, sentenças curtas com um sujeito e objeto, as premissas, com frequência, possuem mais de uma oração.

O método baseado em representação de conhecimento, por outro lado, não se mostrou viável. Apesar de também ter sido utilizado como uma maneira de avaliar a coincidência entre os pares, sua cobertura e acurácia ficaram baixas, principalmente no RTE-3. Em razão disso, optamos por não insistir nesse método.

Nossos métodos, isolados, não seriam suficientes para alcançar performances próximas ou superiores aos modelos avaliados nesses conjuntos de dados. Mesmo quando combinados, classificaram pouco mais da metade dos corpora com resultados modestos. Para além disso, necessitaríamos de análises mais sofisticadas.

Esse dado ficou evidente ao combinarmos o Classificador Bayesiano Ingênuo com o algoritmo Regras de Classificação. O desempenho do classificador foi relativamente melhor apenas quando utilizamos mais um atributo: o tipo de tarefa. Os outros atributos, porém, mais uma vez selecionados com base na coincidência, limitaram a performance do classificador. Em dois dos conjuntos de dados, o RTE-1 e o RTE-2, seus resultados ficaram perto da probabilidade aleatória.

Ao utilizarmos as regras, conseguimos ajustar atributos e parâmetros de modo a ter um maior controle sobre nossos modelos. Nossas regras, ainda que de uma maneira simplificada, nos deram a possibilidade de explorar algumas das características dos pares de forma transparente. Os resultados, porém, mostram que nossos modelos não obtiveram uma performance competitiva tanto em relação a alguns dos trabalhos avaliados nos conjuntos de dados quanto em relação ao modelo baseado em QA implementado com o RoBERTa.

Do ponto de vista dos resultados, somente o método baseado em QA atingiu uma acurácia próxima da classificação humana para os conjuntos de dados e dos

trabalhos avaliados nos desafios. Ainda que pouco transparentes em relação aos parâmetros e atributos necessários para a classificação, modelos como o RoBERTa parecem indicar o caminho para o estado da arte em NLI. Esta parece ser uma constatação frequente nos estudos de processamento linguístico automático hoje em dia: cada vez mais é necessário escolher entre regras transparentes com resultados modestos ou bons resultados obtidos com métodos sem representações simbólicas explícitas.

Referências Bibliográficas

Adams, Rod. "Textual entailment through extended lexical overlap." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 128–133, 2006.

Adams, Rod, Nicolae, Gabriel, Nicolae, Cristina & Harabagiu, Sanda. "Textual entailment through extended lexical overlap and lexico-semantic matching." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 119–124. Prague: Association for Computational Linguistics, 2007. URL https://www.aclweb.org/anthology/W07-1420

Akhmatova, Elena. "Textual entailment resolution via atomic propositions." *In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, vol. 150. Citeseer, 2005.

Andreevskaia, Alina, Li, Zhuoyan & Bergler, Sabine. "Can shallow predicate argument structures determine entailment." *In: Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK*, pp. 45–48, 2005.

Androutsopoulos, Ion & Malakasiotis, Prodromos. "A survey of paraphrasing and textual entailment methods." *Journal of Artificial Intelligence Research*, vol. 38, 135–187, 2010.

Angeli, Gabor, Premkumar, Melvin Jose Johnson & Manning, Christopher D. "Leveraging linguistic structure for open domain information extraction." *In: Proceedings of the 53rd Annual Meeting of the Association for Computational Lin-*

guistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 344–354, 2015.

Baker, Collin F, Fillmore, Charles J & Lowe, John B. "The berkeley framenet project." *In: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pp. 86–90, 1998.

Bar-Haim, Roy, Dagan, Ido, Dolan, Bill, Ferro, Lisa, Giampiccolo, Danilo, Magnini, Bernardo & Szpektor, Idan. "The second pascal recognising textual entailment challenge." *In: Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, vol. 6, pp. 6–4. Venice, 2006.

Bar-Haim, Roy, Dagan, Ido, Greental, Iddo, Szpektor, Idan & Friedman, Moshe. "Semantic inference at the lexical-syntactic level for textual entailment recognition." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 131–136. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1422

Barbosa, Luciano, Cavalin, Paulo R, Guimaraes, Victor & Kormaksson, Matthias. "Methodology and results for the competition on semantic similarity evaluation and entailment recognition for propor 2016." *arXiv preprint arXiv:170908694*, 2017.

Bayer, Samuel, Burger, John, Ferro, Lisa, Henderson, John & Yeh, Alexander. "Mitre's submissions to the eu pascal rte challenge." *In: Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*. Citeseer, 2005.

Bentivogli, Luisa, Cabrio, Elena, Dagan, Ido, Giampiccolo, Danilo, Leggio, Medea Lo & Magnini, Bernardo. "Building textual entailment specialized data sets: a methodology for isolating linguistic phenomena relevant to inference." *In: LREC*, 2010*a*.

Bentivogli, Luisa, Clark, Peter, Dagan, Ido & Giampiccolo, Danilo. "The fifth pascal recognizing textual entailment challenge." *In: TAC*, 2009.

_____. "The sixth pascal recognizing textual entailment challenge." *In: TAC*, 2010*b*.

_____. "The seventh pascal recognizing textual entailment challenge." *In: TAC*, 2011.

Bentivogli, Luisa, Dagan, Ido & Magnini, Bernardo. "The recognizing textual entailment challenges: Datasets and methodologies." *In: Handbook of Linguistic Annotation*, pp. 1119–1147. Springer, 2017.

Blake, Catherine. "The role of sentence structure in recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 101–106. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1417

Bobrow, Daniel, Crouch, Dick, King, Tracy Holloway, Condoravdi, Cleo, Karttunen, Lauri, Nairn, Rowan, de Paiva, Valeria & Zaenen, Annie. "Precision-focused textual inference." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 16–21. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1403

Bos, Johan. "Recognizing textual entailment and computational semantics." *In: Computing meaning*, pp. 89–105. Springer, 2014.

Bos, Johan & Markert, Katja. "Combining shallow and deep nlp methods for recognizing textual entailment." *In: Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK*, pp. 65–68, 2005*a*.

_____. "Recognising textual entailment with logical inference." *In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 628–635, 2005*b*.

_____. "When logical inference helps determining textual entailment (and when it doesn't)." *In: Proceedings of the Second PASCAL RTE Challenge*, p. 26, 2006.

Bowman, Samuel R, Angeli, Gabor, Potts, Christopher & Manning, Christopher D. "A large annotated corpus for learning natural language inference." *arXiv preprint arXiv:150805326*, 2015.

Brysbaert, Marc & New, Boris. "Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english." *Behavior research methods*, vol. 41(4), 977–990, 2009.

Burchardt, Aljoscha & Frank, Anette. "Approaching textual entailment with lfg and framenet frames." *In: Proc. of the Second PASCAL RTE Challenge Workshop.[-]*, 2006.

Burchardt, Aljoscha, Reiter, Nils, Thater, Stefan & Frank, Anette. "A semantic approach to textual entailment: System evaluation and task analysis." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 10–15. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1402

Burek, Gaston, Pietsch, Christian & De Roeck, Anne. "SVO triple based latent semantic analysis for recognising textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 113–118. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1419

Chambers, Nathanael, Cer, Daniel, Grenager, Trond, Hall, David, Kiddon, Chloe, MacCartney, Bill, de Marneffe, Marie-Catherine, Ramage, Daniel, Yeh, Eric & Manning, Christopher D. "Learning alignments and leveraging natural logic." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 165–170. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1427

Chklovski, Timothy & Pantel, Patrick. "Verbocean: Mining the web for fine-grained semantic verb relations." *In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 33–40, 2004.

Clark, Christopher, Lee, Kenton, Chang, Ming-Wei, Kwiatkowski, Tom, Collins, Michael & Toutanova, Kristina. "Boolq: Exploring the surprising difficulty of natural yes/no questions." *arXiv preprint arXiv:190510044*, 2019*a*.

Clark, Kevin, Khandelwal, Urvashi, Levy, Omer & Manning, Christopher D. "What does bert look at? an analysis of bert's attention." *arXiv preprint arXiv:190604341*, 2019*b*.

Clark, Peter, Harrison, Phil, Thompson, John, Murray, William, Hobbs, Jerry & Fellbaum, Christiane. "On the role of lexical and world knowledge in RTE3." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 54–59. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1409

Clarke, Daoud. "Meaning as context and subsequence analysis for entailment." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Condoravdi, Cleo, Crouch, Dick, De Paiva, Valeria, Stolle, Reinhard & Bobrow, Daniel G. "Entailment, intensionality and text understanding." *In: Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pp. 38–45, 2003.

Cooper, Robin, Crouch, Dick, Van Eijck, Jan, Fox, Chris, Van Genabith, Johan, Jaspars, Jan, Kamp, Hans, Milward, David, Pinkal, Manfred, Poesio, Massimo *et al.* "Using the framework." Tech. rep., Technical Report LRE 62-051 D-16, The FraCaS Consortium, 1996.

Crouch, Richard, Karttunen, Lauri & Zaenen, Annie. "Circumscribing is not excluding: A response to manning." *Unpublished manuscript http://www2 parc com/istl/members/karttune/publications/reply-tomanning pdf*, 2006.

Dagan, Ido, Dolan, Bill, Magnini, Bernardo & Roth, Dan. "Recognizing textual entailment: Rational, evaluation and approaches–erratum." *Natural Language Engineering*, vol. 16(1), 105–105, 2010.

Dagan, Ido, Glickman, Oren & Magnini, Bernardo. "The pascal recognising textual entailment challenge." *In: Machine Learning Challenges Workshop*, pp. 177–190. Springer, 2005.

Dagan, Ido, Roth, Dan, Sammons, Mark & Zanzotto, Fabio Massimo. "Recognizing textual entailment: Models and applications." *Synthesis Lectures on Human Language Technologies*, vol. 6(4), 1–220, 2013.

De Marneffe, Marie-Catherine, MacCartney, Bill, Grenager, Trond, Cer, Daniel, Rafferty, Anna & Manning, Christopher D. "Learning to distinguish valid textual entailments." *In: Second Pascal RTE Challenge Workshop*, 2006.

de Salvo Braz, Rodrigo, Girju, Roxana, Punyakanok, Vasin, Roth, Dan & Sammons, Mark. "An inference model for semantic entailment in natural language." *In: Proceedings of the First Challenge Workshop Recognising Textual Entailment*. Springer, 2005.

Delmonte, Rodolfo, Bristot, Antonella, Boniforti, MA Piccolino & Tonelli, Sara. "Coping with semantic uncertainty with venses." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Delmonte, Rodolfo, Bristot, Antonella, Piccolino Boniforti, Marco Aldo & Tonelli, Sara. "Entailment and anaphora resolution in RTE3." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 48–53. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1408

Delmonte, Rodolfo, Tonelli, Sara, Boniforti, Marco Aldo Piccolino & Bristot, Antonella. "Venses–a linguistically-based system for semantic evaluation." *In: Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 49–52, 2005.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton & Toutanova, Kristina. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:181004805*, 2018.

Dzikovska, Myroslava O, Nielsen, Rodney D, Brew, Chris, Leacock, Claudia, Giampiccolo, Danilo, Bentivogli, Luisa, Clark, Peter, Dagan, Ido & Dang, Hoa T. "Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge." Tech. rep., NORTH TEXAS STATE UNIV DENTON, 2013.

Ferrández, Óscar, Micol, Daniel, Muñoz, Rafael & Palomar, Manuel. "A perspective-based approach for solving textual entailment recognition." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 66–71. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1411

Ferrández, Oscar, Muñoz Terol, Rafael, Munoz, Rafael, Martínez-Barco, Patricio, Palomar, Manuel *et al.* "An approach based on logic forms and wordnet relationships to textual entailment performance.", 2006.

Ferreira, M. & Lopes, M. *Para Conhecer Linguística Computacional*. São Paulo: Contexto, 2019.
URL https://books.google.com.br/books?id=N-EZyAEACAAJ

Ferrés, Daniel & Rodríguez, Horacio. "Machine learning with semantic-based distances between sentences for textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 60–65. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1410

Fialho, Pedro, Marques, Ricardo, Martins, Bruno, Coheur, Luisa & Quaresma, Paulo. "Inesc-id at assin:: measuring semantic similarity and recognizing textual entailment." *Linguamática*, vol. 8(2), 33–42, 2016.

Fonseca, E, Santos, L, Criscuolo, Marcelo & Aluisio, S. "Assin: Avaliacao de similaridade semantica e inferencia textual." *In: Computational Processing*

of the Portuguese Language-12th International Conference, Tomar, Portugal, pp. 13–15, 2016.

Fonseca, Erick Rocha. *Reconhecimento de implicaçao textual em português*. Ph.D. thesis, Universidade de São Paulo, 2018.

Fowler, Abraham, Hauser, Bob, Hodges, Daniel, Niles, Ian, Novischi, Adrian & Stephan, Jens. "Applying cogex to recognize textual entailment." *In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 69–72. Citeseer, 2005.

Garoufi, Konstantina. *Towards a better understanding of applied textual entailment*. Ph.D. thesis, Citeseer, 2007.

Giampiccolo, Danilo, Dang, Hoa Trang, Magnini, Bernardo, Dagan, Ido, Cabrio, Elena & Dolan, Bill. "The fourth pascal recognizing textual entailment challenge." *In: TAC*. Citeseer, 2008.

Giampiccolo, Danilo, Magnini, Bernardo, Dagan, Ido & Dolan, Bill. "The third PASCAL recognizing textual entailment challenge." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1401

Glickman, Oren, Dagan, Ido & Koppel, Moshe. "Web based probabilistic textual entailment." *In: Proceedings of the 1st Pascal Challenge Workshop*, 2005.

Grice, Herbert P. "Logic and conversation." *In: Speech acts*, pp. 41–58. Brill, 1975.

Habash, Nizar & Dorr, Bonnie. "Catvar: A database of categorial variations for english." *In: Proceedings of the MT Summit*, pp. 471–474. Citeseer, 2003.

Harmeling, Stefan. "An extensible probabilistic transformation-based approach to the third recognizing textual entailment challenge." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 137–142. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1423

Harris, Zellig S. "Distributional structure." *Word*, vol. 10(2-3), 146–162, 1954.

Herrera, Jesús, Penas, Anselmo, Rodrigo, Alvaro & Verdejo, Felisa. "Uned at pascal rte-2 challenge." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Herrera, Jesús, Penas, Anselmo & Verdejo, Felisa. "Textual entailment recognition based on dependency analysis and wordnet." *In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 21–24. Springer, 2005.

Hickl, Andrew & Bensley, Jeremy. "A discourse commitment-based framework for recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 171–176. Prague: Association for Computational Linguistics, 2007*a*.
URL https://www.aclweb.org/anthology/W07-1428

_____. "A discourse commitment-based framework for recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 171–176, 2007*b*.

Hickl, Andrew, Williams, John, Bensley, Jeremy, Roberts, Kirk, Rink, Bryan & Shi, Ying. "Recognizing textual entailment with lcc's groundhog system." *In: Proceedings of the Second PASCAL Challenges Workshop*, vol. 18, 2006.

Iftene, Adrian & Balahur-Dobrescu, Alexandra. "Hypothesis transformation and semantic variability rules used in recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 125–130. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1421

Inkpen, Diana, Kipp, Darren & Nastase, Vivi. "Machine learning experiments for textual entailment." *In: Proceedings of the Second Challenge Workshop Recognising Textual Entailment*, pp. 17–20, 2006.

Jacobson, Pauline I. *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford University Press, 2014.

Jia, Houping, Huang, Xiaojiang, Ma, Tengfei, Wan, Xiaojun & Xiao, Jianguo. "Pkutm participation at tac 2010 rte and summarization track." *In: In Proceedings of Text Analysis Conference (TAC*. Citeseer, 2010.

Jijkoun, Valentin, de Rijke, Maarten *et al.* "Recognizing textual entailment using lexical similarity." *In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 73–76. Citeseer, 2005.

Jurafsky, D & Martin, JH. "Naive bayes and sentiment classification." *Speech and Language Processing*, vol. 7, 2019.

Katrenko, Sophia, Adriaans, Peter *et al.* "Using maximal embedded syntactic subtrees for textual entailment recognition." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Khot, Tushar, Sabharwal, Ashish & Clark, Peter. "Scitail: A textual entailment dataset from science question answering." *In: AAAI*, vol. 17, pp. 41–42, 2018.

Kingma, Diederik P & Ba, Jimmy. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:14126980*, 2014.

Korman, Daniel Z, Mack, Eric, Jett, Jacob & Renear, Allen H. "Defining textual entailment." *Journal of the Association for Information Science and Technology*, vol. 69(6), 763–772, 2018.

Kouylekov, Milen & Magnini, Bernardo. "Recognizing textual entailment with tree edit distance algorithms." *In: Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pp. 17–20, 2005.

_____. "Tree edit distance for recognizing textual entailment: Estimating the cost of insertion." *In: Proc. of the PASCAL RTE-2 Challenge*, pp. 68–73, 2006.

Kouylekov, Milen & Negri, Matteo. "An open-source package for recognizing textual entailment." *In: Proceedings of the ACL 2010 System Demonstrations*, pp. 42–47, 2010.

Kozareva, Zornitsa & Montoyo, Andrés. "Mlent: The machine learning entailment system of the university of alicante." *In: Proc. of 2nd PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Levy, Omer, Zesch, Torsten, Dagan, Ido & Gurevych, Iryna. "Ukp-biu: Similarity and entailment metrics for student response analysis." *In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 285–289, 2013.

Li, Baoli, Irwin, Joseph, Garcia, Ernest V. & Ram, Ashwin. "Machine learning based semantic inference: Experiments and observations at RTE-3." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 159–164. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1426

Lin, Dekang *et al.* "An information-theoretic definition of similarity." *In: Icml*, vol. 98, pp. 296–304. Citeseer, 1998.

Litkowski, Ken. "Componential analysis for recognizing textual entailment." *In: The Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Liu, Xiaodong, He, Pengcheng, Chen, Weizhu & Gao, Jianfeng. "Multi-task deep neural networks for natural language understanding." *arXiv preprint arXiv:190111504*, 2019*a*.

Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke & Stoyanov, Veselin. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:190711692*, 2019*b*.

MacCartney, Bill. *Natural Language Inference*. Ph.D. thesis, Stanford University, 2009.

Malakasiotis, Prodromos & Androutsopoulos, Ion. "Learning textual entailment using SVMs and string similarity measures." *In: Proceedings of the*

ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 42–47. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1407

Manning, Christopher & Schutze, Hinrich. *Foundations of statistical natural language processing*. MIT press, 1999.

Manning, Christopher D. "Local textual inference: it's hard to circumscribe, but you know it when you see it–and nlp needs it.", 2006.

Marelli, Marco, Bentivogli, Luisa, Baroni, Marco, Bernardi, Raffaella, Menini, Stefano & Zamparelli, Roberto. "Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment." *In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 1–8, 2014.

Marsi, Erwin, Krahmer, Emiel & Bosma, Wauter. "Dependency-based paraphrasing for recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 83–88. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1414

Marsi, Erwin, Krahmer, Emiel, Bosma, Wauter & Theune, Mariët. "Normalized alignment of dependency trees for detecting textual entailment." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 56–61, 2006.

Martinez-Gómez, Pascual, Mineshima, Koji, Miyao, Yusuke & Bekki, Daisuke. "ccg2lambda: A compositional semantics system." *In: Proceedings of ACL-2016 System Demonstrations*, pp. 85–90, 2016.

McHugh, Mary L. "Interrater reliability: the kappa statistic." *Biochemia medica: Biochemia medica*, vol. 22(3), 276–282, 2012.

Miller, George A. "Wordnet: a lexical database for english." *Communications of the ACM*, vol. 38(11), 39–41, 1995.

Mohammad, Saif, Dorr, Bonnie J, Egan, Melissa, Madnani, Nitin, Zajic, David M & Lin, Jimmy J. "Multiple alternative sentence compressions and word-pair antonymy for automatic text summarization and recognizing textual entailment." *In: TAC*, 2008.

Montejo-Ráez, Arturo, Perea, Jose Manuel, Martínez-Santiago, Fernando, García-Cumbreras, Miguel Ángel, Martín-Valdivia, Maite & Ureña-López, Alfonso. "Combining lexical-syntactic information with machine learning for recognizing textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 78–82. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1413

Nadeau, David & Sekine, Satoshi. "A survey of named entity recognition and classification." *Lingvisticae Investigationes*, vol. 30(1), 3–26, 2007.

Nangia, Nikita, Williams, Adina, Lazaridou, Angeliki & Bowman, Samuel R. "The repeval 2017 shared task: Multi-genre natural language inference with sentence representations." *arXiv preprint arXiv:170708172*, 2017.

Newman, Eamonn, Dunnion, John & Carthy, Joe. "Constructing a decision tree classifier using lexical and syntactic features." *In: Proc. of 2nd PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Newman, Eamonn, Stokes, Nicola, Dunnion, John & Carthy, Joe. "Ucd iirg approach to the textual entailment challenge." *In: Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*, 2005.

Nicholson, Jeremy, Stokes, Nicola & Baldwin, Timothy. "Detecting entailment using an extended implementation of the basic elements overlap metrics." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 122–127, 2006.

Nielsen, R, Ward, Wayne & Martin, James H. "Toward dependency path based entailment." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 44–49, 2006.

Ott, Niels, Ziai, Ramon, Hahn, Michael & Meurers, Detmar. "Comet: Integrating different levels of linguistic modeling for meaning assessment." *In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 608–616, 2013.

Paiva, Valeria de, Rademaker, Alexandre & Melo, Gerard de. "Openwordnet-pt: An open brazilian wordnet for reasoning." Tech. rep., COLING 2012, 2012.

Pazienza, Maria Teresa, Pennacchiotti, Marco & Zanzotto, Fabio Massimo. "Textual entailment as syntactic graph distance: a rule based and a svm based approach." *In: Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*, 2005.

Pérez, Diana & Alfonseca, Enrique. "Application of the bleu algorithm for recognising textual entailments." *In: Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pp. 9–12. Citeseer, 2005.

Raina, Rajat, Haghighi, Aria, Cox, Christopher, Finkel, Jenny, Michels, Jeff, Toutanova, Kristina, MacCartney, Bill, de Marneffe, Marie-Catherine, Manning, Christopher D & Ng, Andrew Y. "Robust textual inference using diverse knowledge sources." *In: Proc. of the 1st. PASCAL Recognision Textual Entailment Challenge Workshop, Southampton, UK*, pp. 57–60, 2005.

Real, Livy, Fonseca, Erick & Oliveira, Hugo Gonçalo. "The assin 2 shared task: a quick overview." *In: International Conference on Computational Processing of the Portuguese Language*, pp. 406–412. Springer, 2020.

Real, Livy, Rodrigues, Ana, e Silva, Andressa Vieira, Albiero, Beatriz, Thalenberg, Bruna, Guide, Bruno, Silva, Cindy, de Oliveira Lima, Guilherme, Câmara, Igor CS, Stanojević, Miloš *et al.* "Sick-br: a portuguese corpus for inference." *In: International Conference on Computational Processing of the Portuguese Language*, pp. 303–312. Springer, 2018.

Rodrigo, Álvaro, Peñas, Anselmo, Herrera, Jesús & Verdejo, Felisa. "Experiments of UNED at the third recognising textual entailment challenge." *In:*

Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 89–94. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1415

Roth, Dan & Sammons, Mark. "Semantic and logical inference model for textual entailment." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 107–112. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1418

Rus, Vasile. "Two related lexico-syntactic approaches to entailment." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Sammons, Mark. *17 Recognizing Textual Entailment*. Wiley Online Library, 2015.

Schilder, Frank & McInnes, Bridget Thomson. "Word and tree-based similarities for textual entailment." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 140–145, 2006.

Settembre, Scott. "Textual entailment using univariate density model and maximizing discriminant function." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 95–100. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1416

Tan, Chuanqi, Sun, Fuchun, Kong, Tao, Zhang, Wenchang, Yang, Chao & Liu, Chunfang. "A survey on deep transfer learning." *In: International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Tatu, Marta, Iles, Brandon, Slavick, John, Novischi, Adrian & Moldovan, Dan. "Cogex at the second recognizing textual entailment challenge." *In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 104–109. Citeseer, 2006.

Tatu, Marta & Moldovan, Dan. "Cogex at rte 3." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 22–27. Prague:

Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1404

Tsuchida, Masaaki & Ishikawa, Kai. "Ikoma at tac2011: A method for recognizing textual entailment using lexical-level and sentence structure-level features." *In: TAC*, 2011.

Vanderwende, Lucy & Dolan, William B. "What syntax can contribute in the entailment task." *In: Machine Learning Challenges Workshop*, pp. 205–216. Springer, 2005.

Vanderwende, Lucy, Menezes, Arul & Snow, Rion. "Microsoft research at rte-2: Syntactic contributions in the entailment task: an implementation." *In: Proceedings of the Second PASCAL Challenges Workshop*, 2006.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz & Polosukhin, Illia. "Attention is all you need." *In: Advances in neural information processing systems*, pp. 5998–6008, 2017.

Voorhees, Ellen M. "Contradictions and justifications: Extensions to the textual entailment task." *In: Proceedings of ACL-08: HLT*, pp. 63–71, 2008.

Wang, Rui & Neumann, Günter. "Recognizing textual entailment using sentence similarity based on dependency tree skeletons." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 36–41. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1406

Wang, Rui, Zhang, Yi & Neumann, Guenter. "A joint syntactic-semantic representation for recognizing textual relatedness." *In: TAC*, 2009.

Williams, Adina, Nangia, Nikita & Bowman, Samuel R. "A broad-coverage challenge corpus for sentence understanding through inference." *arXiv preprint arXiv:170405426*, 2017.

Wu, Dekai. "Textual entailment recognition based on inversion transduction grammars." *In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 37–40. Citeseer, 2005.

Wu, Fei & Weld, Daniel S. "Open information extraction using wikipedia." *In: Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 118–127, 2010.

Yoshikawa, Masashi, Noji, Hiroshi & Matsumoto, Yuji. "A* CCG parsing with a supertag and dependency factored model." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 277–287, 2017.

Zaenen, Annie, Karttunen, Lauri & Crouch, Richard. "Local textual inference: can it be defined or circumscribed?" *In: Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pp. 31–36. Association for Computational Linguistics, 2005.

Zanzotto, Fabio Massimo, Pennacchiotti, Marco & Moschitti, Alessandro. "Shallow semantic in fast textual entailment rule learners." *In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 72–77. Prague: Association for Computational Linguistics, 2007.
URL https://www.aclweb.org/anthology/W07-1412

Zanzotto, FM, Moschitti, Alessandro, Pennacchiotti, Marco & Pazienza, MT. "Learning textual entailment from examples." *In: Second PASCAL recognizing textual entailment challenge*, p. 50. PASCAL, 2006.

Zhu, Yukun, Kiros, Ryan, Zemel, Rich, Salakhutdinov, Ruslan, Urtasun, Raquel, Torralba, Antonio & Fidler, Sanja. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books." *In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.