UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESCREVENDO REGIÕES DE IMAGENS ATRAVÉS DE REDES NEURAIS PROFUNDAS E *ABSTRACT MEANING REPRESENTATION*

ANTONIO MANOEL DOS SANTOS ALMEIDA NETO

ORIENTADORA: PROFA. DRA. HELENA DE MEDEIROS CASELI

CO-ORIENTADOR: PROF. DR. TIAGO AGOSTINHO DE ALMEIDA

São Carlos – SP

Maio, 2020

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESCREVENDO REGIÕES DE IMAGENS ATRAVÉS DE REDES NEURAIS PROFUNDAS E *ABSTRACT MEANING REPRESENTATION*

ANTONIO MANOEL DOS SANTOS ALMEIDA NETO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.

Orientadora: Profa. Dra. Helena de Medeiros Caseli

São Carlos – SP

Maio, 2020

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Antônio Manoel dos Santos Almeida Neto, realizada em 28/05/2020:

Helena de M. Caseli

Profa. Dra. Helena de Medeiros Caseli
UFSCar

Prof. Dr. Ricardo Cerri
UFSCar

Prof. Dr. Ivandré Paraboni
USP

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Ricardo Cerri, Ivandré Paraboni e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Helena de M. Caseli

Profa. Dra. Helena de Medeiros Caseli

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus por me guiar, iluminar e tranquilizar nos momentos necessários, me ajudando a seguir em frente com meus objetivos, mesmo diante das dificuldades.

À toda minha família, em especial minha irmã Andresa, meus sobrinhos Gabriel e Miguel, minha mãe Sofia e ao meu padrasto Nelson, por todo o amor e incentivo nessa minha caminhada.

À professora Helena por toda a orientação, e também ao professor Tiago pelas dicas, discussões e apoio com a máquina para realização dos experimentos. Sem a ajuda de vocês a realização deste trabalho não seria possível.

Aos meus amigos da pós graduação e do LALIC, por todas as conversas, cafés, momentos de lazer e diversão que tivemos juntos.

Ao programa de Pós-Graduação da Universidade Federal de São Carlos, professores e funcionários pelo suporte para o desenvolvimento deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo auxílio financeiro durante a realização.

A todos que auxiliaram de alguma forma, diretamente ou indiretamente na realização deste trabalho, meu muito obrigado!

RESUMO

O mundo que nos cerca é composto por imagens que, muitas vezes, precisam ser traduzidas em palavras. Essa tradução pode se dar em partes, convertendo regiões da imagem em descrições textuais. A descrição da região de uma imagem é a transformação da informação contida nesta área para palavras em língua natural, de modo a expressar a maneira como os objetos se relacionam entre si. Recentemente, modelos computacionais que procuram desempenhar essa tarefa de maneira semelhante aos seres humanos estão sendo propostos, principalmente utilizando redes neurais profundas (*deep learning*). Como forma de melhorar a qualidade das sentenças produzidas por um desses modelos, este trabalho verificou a empregabilidade da representação semântica *Abstract Meaning Representation* (AMR) na geração de descrições para regiões de imagem. A AMR foi investigada como formalismo de representação, em alternativa à língua natural, empregando-a com algumas variações, para que o modelo de aprendizado de máquina, utilizando redes neurais profundas, fosse capaz de prever sentenças em tal representação. A hipótese deste trabalho, de que a utilização de sentenças em forma de AMR resultaria em melhores descrições foi confirmada parcialmente, visto que o modelo treinado com AMR foi superior em quase todas as avaliações.

Palavras-chave: descrição de regiões de imagem. representação semântica. *abstract meaning representation*. *dense captioning*.

ABSTRACT

The world around us is composed of images that often need to be translated into words. This translation can take place in parts, converting regions of the image into textual descriptions. The description of the region of an image is the transformation of the information contained in this area into words in natural language, to express the way objects relate to each other. Recently, computational models that seek to perform this task in a similar way to human beings are being proposed, mainly using deep neural networks. As a way to improve the quality of the sentences produced by one of these models, this work verified the employability of the Abstract Meaning Representation (AMR) semantic representation in the generation of descriptions for image regions. AMR was investigated as representation formalism, as an alternative to natural language, using it with some variations, so that the machine learning model, using deep neural networks, was able to predict sentences in such representation. The hypothesis of this study, that the use of sentences in the form of AMR would result in better descriptions, was partially confirmed, since the model trained with AMR was superior in almost all evaluations.

Keywords: description of image regions. semantic representation. abstract meaning representation. dense captioning.

LISTA DE SIGLAS

AMR	*Abstract Meaning Representation*
AMT	*Amazon Mechanical Turk*
AP	*Average Precision*
bbox	*bouding box*
BLEU	*Bilingual Evaluation Understudy*
BP	*Brevity Penalty*
BRNN	*Bidirectional Recurrent Neural Network*
CAG-Net	*Context and Attribute Grounded Dense Captioning*
CNN	*Convolution Neural Network*
FCLN	*Fully Convolutional Localization Network*
FN	Falsos Negativos
FP	Falsos Positivos
GDRI	Geração de Descrições para Regiões de Imagens
GER	Geração de Expressões de Referência
GLN	Geração de Língua Natural
GPU	*Graphics Processing Unit*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
IoU	*Intersection over Union*
LSTM	*Long Short Term Memory*
mAP	*Mean Average Precision*
METEOR	*Metric for Evaluation of Translation with Explicit Ordering*

MLP	*Multilayer Perceptron*
MM	*Max-Margin*
MMI	*Maximum Mutual Information*
MRNN	*Multimodal Recurrent Neural Network*
MSCOCO	*Microsoft Common Object in Context*
NER	*Named Entity Recognizer*
PFE	*Precise Feature Extraction*
PLN	Processamento de Língua Natural
RCNN	*Region with CNN features*
RNA	Redes Neurais Artificiais
RNN	*Recurrent Neural Network*
ROUGE	*Recall-Oriented Understudy for Gisting*
RPN	*Region Proposal Network*
RS	Representação Semântica
RoI	*Region of Interest*
SMATCH	*Semantic Match*
TAC	*Text Analysis Conference*
VG	*Visual Genome*
VP	Verdadeiros Positivos
YFCC100M	*Yahoo Flickr Creative Commons 100 Million*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1
INTRODUÇÃO

O processamento automático de línguas naturais é a vertente da inteligência artificial que procura automatizar e reproduzir o comportamento humano a respeito do conhecimento linguístico específico sobre um determinado idioma.

Dentro do processamento de línguas naturais, a geração de texto em língua natural começou a atrair grande interesse de pesquisadores desde a segunda guerra mundial, dada a necessidade de gerar tradução automática entre idiomas (JURAFSKY; MARTIN, 2018; GARCÍA-MÉNDEZ et al., 2019). A geração de línguas naturais tem como finalidade produzir alguma representação final, ou parcial, de um determinado idioma (REITER; DALE, 1997). Contudo, os resultados na época não foram compatíveis com a grande expectativa criada e investimentos realizados. Como consequência, a área, assim como todas pertencentes à grande área de inteligência artificial, sofreu com a escassez de recursos nas décadas seguintes.

Entre as aplicações que demandam a geração de língua natural, está a **Geração de Descrições para Regiões de Imagens** (GDRI). Essa descrição é a transformação da informação, que está na forma de imagem, em uma informação equivalente em formato de texto, em alguma língua natural. Segundo Zhang et al. (2015), a geração de descrições para regiões de imagens é definida como a produção de uma sentença que expressa a maneira como os objetos se relacionam entre si e seus atributos em uma determinada região em uma imagem.

Existem duas tarefas que são comumente confundidas com a geração de descrições para regiões de imagens: a classificação automática de objetos e a geração automática de legenda de imagens. A classificação de objetos é o processo de rotulação, a partir de um conjunto pré-definido de palavras (rótulos), de um determinado objeto identificado em uma imagem. Essa tarefa difere da GDRI visto que apenas a classificação do objeto pode ser insuficiente para a mesma informação estar presente em forma de língua natural e por não relacionar os objetos da imagem.

A outra tarefa que é comumente relacionada à GDRI é a de geração automática de legenda de imagens. A geração de legenda de imagens é o processo de atribuição de uma única sentença para a imagem completa, ou seja, uma frase que tem como finalidade descrever a

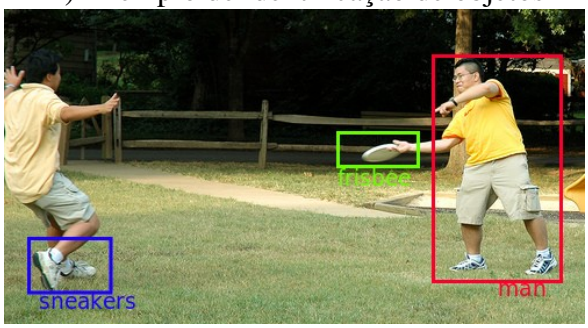
imagem por completo, e não apenas regiões específicas. Uma imagem contém uma legenda, mas da mesma forma, possui várias regiões, às quais, na GDRI, uma sentença será associada.

Para ilustrar a diferença entre essas três tarefas, considere a Figura 1. A parte A da figura ilustra o uso de uma única sentença (legenda) com o intuito de descrever a imagem por completo. Na parte B ocorre a identificação de objetos, rotulando-os, e por último a parte C demonstra a descrição para algumas regiões da imagem, inclusive relacionando os objetos entre si.

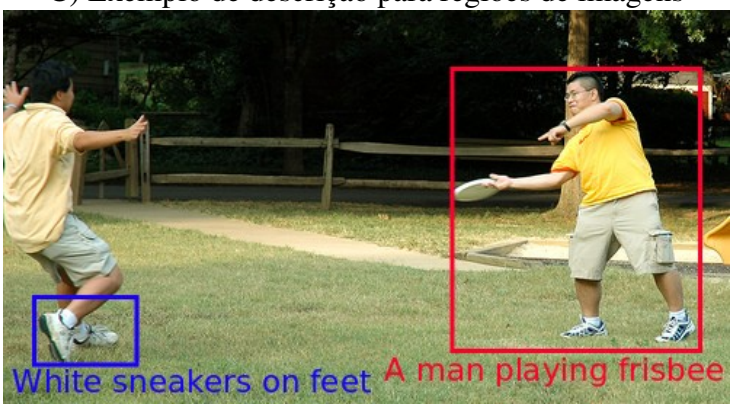
Figura 1 – Diferenciação de tarefas similares a GDRI

A) Exemplo de legenda para imagem

B) Exemplo de identificação de objetos



C) Exemplo de descrição para regiões de imagens



Fonte: Adaptado do conjunto *Visual Genome* de Krishna et al. (2017)

A geração de descrições para regiões de imagens é realizada por nós, seres humanos, quando descrevemos (geralmente na forma verbal) uma cena a partir de tudo o que a nossa visão captura. Nesse sentido, a GDRI é a descrição de uma pequena parte do que estamos vendo. Embora essa seja uma tarefa relativamente simples de ser realizada por um humano, para a computação, trata-se de uma lacuna parcialmente resolvida e para a qual foram propostos meios para reproduzir essa tarefa de forma eficiente, ou próxima à realizada por humanos.

Como forma de investigação de uma possível melhora na qualidade das descrições para regiões de imagens por meios computacionais, neste trabalho, são empregadas informações de nível semântico da língua. De acordo com Specia e Rino (2002), a semântica diz respeito ao significado, o sentido que deseja-se transmitir, independente das palavras e ordem utilizadas. Desta forma, a semântica é capaz de identificar os sentidos, independente da forma morfológica

e sintática. A representação semântica escolhida foi a *Abstract Meaning Representation* de Banarescu et al. (2013). Atualmente, é a representação semântica que mais tem atraído interesse na literatura em diversas tarefas, como a tradução (TAMCHYNA et al., 2015) e sumarização automática (LIAO et al., 2018). Como um dos seus principais benefícios, cita-se a a unificação de várias tarefas semânticas (entidades nomeadas, resolução de co-referências entre outras)

As próximas seções apresentam os pontos específicos deste trabalho no que se refere à sua justificativa, motivação, objetivo e hipótese.

1.1 Justificativa e motivação

Os métodos mais utilizados, atualmente, para a geração de descrições para regiões de imagem baseiam-se em aprendizado de máquina e têm como entrada e saída a língua natural. Simplificadamente, a partir de exemplos de sentenças em língua natural, e suas respectivas regiões da imagem, tais métodos aprendem a gerar descrições também em língua natural. Essa abordagem de utilizar a língua natural como entrada e saída do modelo é comum, assim como em outras áreas do processamento de línguas naturais, como na recuperação de informação, tradução e sumarização automática de textos. No entanto, Karpathy et al. (2014) demonstram que a utilização de elementos sintáticos, sobretudo de relações de dependência, beneficiam a tarefa de recuperação de informação, quando comparadas ao uso da língua natural de maneira pura (sem pré-processamento ou enriquecimento de qualquer espécie).

A conclusão de Karpathy et al. (2014) motiva a investigação deste projeto a respeito de uso de elementos de outros níveis da língua, mais especificamente do nível semântico, para favorecer modelos computacionais que visam a geração automática de descrições para regiões de imagem, em contra-partida à abordagem convencional de utilização de descrições em língua natural.

Como exemplo do poder de abstração que a AMR é capaz de prover, têm-se as seguintes sentenças, retiradas do trabalho de Anchiêta e Pardo (2018), escritas de formas diferentes, mas que expressam o mesmo sentido. A Figura 2 ilustra a representação AMR para as 3 sentenças. Como as sentenças tem o mesmo significado, de que “a garota ajustou a máquina”, são representadas de uma única forma em AMR.

The girl made adjustment to the machine.

The girl adjusted the machine.

The machine was adjusted by the girl.

Figura 2 – Exemplo AMR

```
(a / adjust-01
      :ARG0 (g / girl)
      :ARG1 (m / machine))
```

Fonte: Retirado de Anchiêta e Pardo (2018)

1.2 Objetivo

O objetivo deste trabalho é verificar se a representação semântica *Abstract Meaning Representation* é capaz de auxiliar o processo de geração de descrições para regiões de imagens.

Para atingir esse objetivo, foram utilizados modelos e algoritmos de aprendizado de máquina, em especial as redes neurais artificiais profundas, em virtude de serem as técnicas com melhores resultados na atualidade, de acordo com a literatura.

1.3 Hipótese

Este trabalho parte da hipótese de que a representação semântica *Abstract Meaning Representation* pode auxiliar o processo de geração de descrições de regiões de imagem preservando melhor o conteúdo semântico do que o uso de língua natural pura. Até onde se tem conhecimento, este é o primeiro trabalho que se propõe a utilizar AMR para a GDRI.

Para validar essa hipótese, foram realizados experimentos com formas variadas de uso da *Abstract Meaning Representation*, comparado-as com os resultados obtidos com a abordagem tradicional de usar língua natural pura, ou seja, tradicionalmente empregada na literatura.

1.4 Organização do trabalho

Para melhor entendimento do desenvolvimento deste trabalho, os próximos capítulos estão estruturados na seguinte forma:

- O Capítulo 2 oferece o embasamento teórico necessário para a compreensão das técnicas utilizadas no desenvolvimento deste trabalho.
- O Capítulo 3 define o processamento de línguas naturais e tem como foco principal a subárea de interesse neste trabalho: a geração de língua natural. Para tanto, além da geração de descrições para regiões de imagem, são apresentadas as aplicações relacionadas, como a geração de legendas para imagens e outras que fizeram uso da *Abstract Meaning Representation* (AMR) para diferentes tarefas. Por último, são apresentadas as medidas de avaliação geralmente empregadas para avaliar a geração de língua natural dentro do escopo deste projeto, para posterior análise dos modelos propostos.

- O Capítulo 4 apresenta os principais trabalhos relacionados à geração de descrições para regiões de imagens e aplicações correlatas, seus modelos e principais características. Por último, apresenta-se uma tabela com a sumarização dos principais pontos de cada um dos trabalhos descritos nesse capítulo.

- O Capítulo 5 descreve o método desenvolvido neste trabalho para realizar a descrição de regiões da imagem usando AMR.

- O Capítulo 6 apresenta o conjunto de dados empregado neste trabalho, assim como os experimentos realizados, os resultados obtidos, os parâmetros para os processamentos, e as formas de avaliação.

- E por fim, o Capítulo 7 traz as considerações finais, fazendo uma síntese dos pontos principais deste trabalho, apresentando também os trabalhos futuros que podem ser desenvolvidos a partir deste trabalho, e elencando suas contribuições.

Capítulo 2
FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve a fundamentação teórica deste projeto. Assim, a Seção 2.1 apresenta os fundamentos para se entender a geração de descrições de regiões da imagem usando redes neurais artificiais. A Seção 2.2, por sua vez, descreve o formalismo semântico adotado neste trabalho.

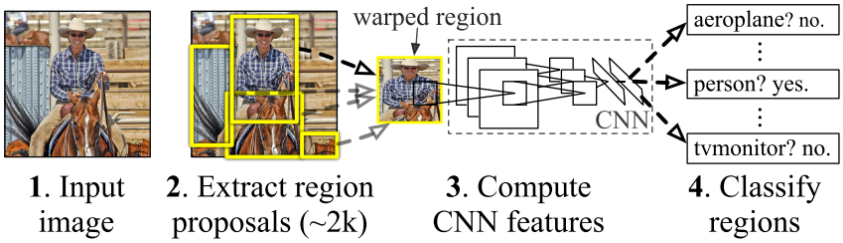
2.1 Redes neurais artificiais

Com o aumento do uso de técnicas de Redes Neurais Artificiais (RNA) profundas (*deep learning*), tais estratégias também começaram a serem utilizadas para a tarefa de geração de língua natural. Da mesma forma, a tarefa de geração de descrições para regiões de imagem também faz uso de tais técnicas, como será detalhado no Capítulo 4.

De modo geral, as principais técnicas podem ser divididas em duas etapas: o reconhecimento de objetos (Seção 2.1.1), responsável por realizar o tratamento da imagem, e a geração da língua natural efetivamente (Seção 2.1.2), em que recebe a entrada da etapa anterior.

2.1.1 Arquiteturas para identificação de objetos

Figura 3 – Arquitetura RCNN



Fonte: Retirado de Girshick et al. (2014)

A primeira arquitetura utilizada pelos trabalhos relacionados (Capítulo 4) para identificação de objetos é a *Region with CNN features* (RCNN), proposta por Girshick et al. (2014).

Conforme ilustra a Figura 3, a arquitetura é composta por três módulos: a proposta de regiões, a extração de características e a classificação de objetos. O nome da arquitetura se deve ao fato de utilizar-se a *Convolution Neural Network* (CNN - Rede Neural Artificial Convolucional), que é responsável apenas pela extração das características, não realizando a classificação propriamente, sendo necessária a utilização de outras técnicas para esta finalidade.

No primeiro módulo de proposta de regiões, ocorre a identificação e proposta de regiões a partir de uma imagem, sendo essas regiões objetos únicos ou combinação de vários objetos. Dentre as várias técnicas que podem ser utilizadas, os autores utilizaram o *selective search* de Uijlings et al. (2013).

O segundo módulo faz a extração de características para as regiões propostas pelo módulo anterior. A arquitetura utiliza a rede CNN para realizar a extração, tendo como entrada o tamanho de 227 x 227, com um vetor de 4096 dimensões e cinco camadas convolucionais. O treinamento ocorreu utilizando a *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) de 2012 (RUSSAKOVSKY et al., 2015).

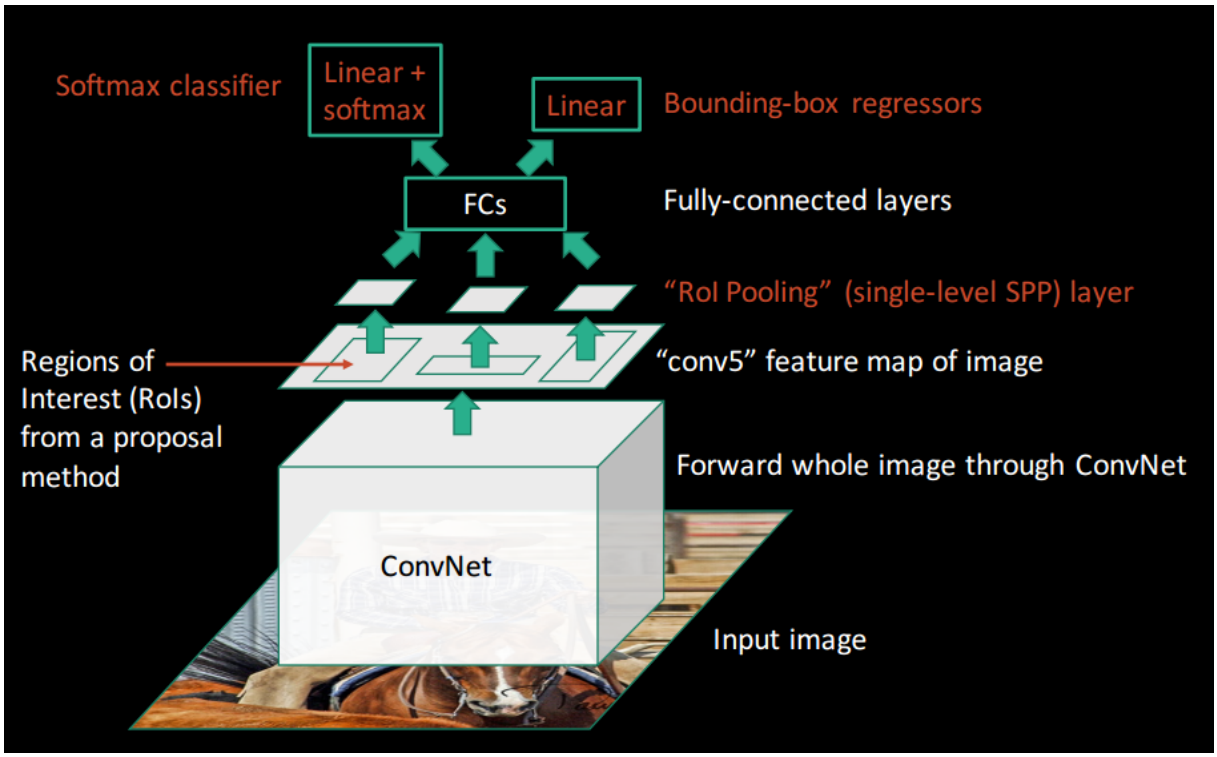
O terceiro módulo realiza a classificação dos objetos. A partir do conjunto de características resultantes da extração da CNN, é realizada a classificação utilizando o *Support Vector Machines* (CORTES; VAPNIK, 1995), para que então o objeto possa ser rotulado.

Um passo a mais realizado pela arquitetura é a regressão da *bouding box* (bbox). Essa regressão tem como finalidade melhorar a *bbox*, isto é, corrigir erros gerados pelo módulo de proposta de regiões, como um ajuste fino. Esse ajuste é realizado através de uma regressão linear. Com essa *bbox* ajustada sendo a entrada da rede convolucional (descartando a proveniente do *selective search*), houve uma melhora no resultado na classificação dos objetos no terceiro módulo.

Apesar desta arquitetura representar um avanço na detecção de objetos, existem alguns problemas importantes a ressaltar. O primeiro é que a regressão da *bbox* não é aprendida, ou seja, a arquitetura não aprende a gerar uma *bbox* melhor, sendo necessária a execução do modelo novamente, agora com a *bbox* ajustada e não com a proveniente do *selective search*. Outro problema é o tempo de execução, cerca de 50 segundos para cada imagem. Buscando resolver esses problemas, Girshick (2015) propôs a *Fast RCNN*. Essa arquitetura busca melhorar a performance da RCNN por meio de algumas alterações na arquitetura.

Na *Fast RCNN*, conforme ilustra a Figura 4, a CNN recebe como entrada a imagem completa e suas respectivas regiões selecionadas com o *selective search*, e não apenas uma região por vez. Essa alteração permite que a extração de característica ocorra uma única vez, acelerando a performance. Dessa forma o método de regiões de interesse (*Region of Interest* - RoI) converte as características em um mapa de características menor, que será a entrada para as camadas totalmente conectadas. Por fim, o resultado das camadas conectadas é utilizado para realizar a classificação, agora através da *softmax*, e a regressão da *bbox*.

Figura 4 – Arquitetura Fast RCNN



Fonte: Retirado de http://www.robots.ox.ac.uk/ tvg/publications/talks/fast-rcnn-slides.pdf

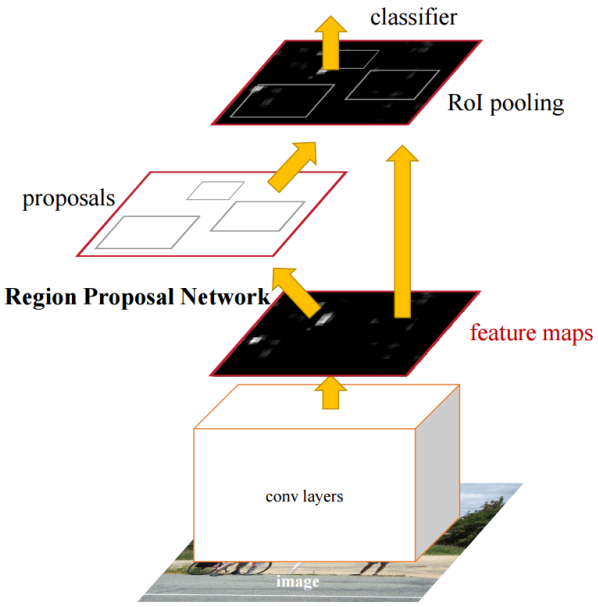
Apesar das melhorias, a extração das regiões através do *selective search* é considerada lenta (cerca de 2 segundos por imagem), inviabilizando a utilização em aplicações de tempo real. Para contornar esse problema surgiu uma outra variação da arquitetura, a *Faster RCNN*, proposta por Ren et al. (2017).

A ideia da *Faster RCNN* é eliminar a utilização do método *selective search* para melhorar o desempenho. Conforme ilustra a Figura 5, é utilizada a Rede de Proposta de Região (*Region Proposal Network* - RPN) ao final da extração de característica da CNN. O classificador utiliza as regiões propostas pela RPN e a RoI. Com essas alterações, a *Faster RCNN* diminuiu significativamente o tempo de execução (para 0,2 segundos) e ainda viabilizou a arquitetura a aprender a *bbox* ajustada pela regressão.

2.1.2 Redes neurais recorrentes

As Redes Neurais Recorrentes (*Recurrent Neural Network* - RNN) são utilizadas em aplicações em que existe a necessidade de considerar a sequência atual em relação às anteriores durante um tempo, ou período, como em aplicações de áudio e sinais. Essa noção de tempo não se resume a apenas tempo no sentido literal da palavra, mas também pode ser aplicada onde há a necessidade de processar informações de maneira sequencial, como em um texto formado por uma sequência de palavras.

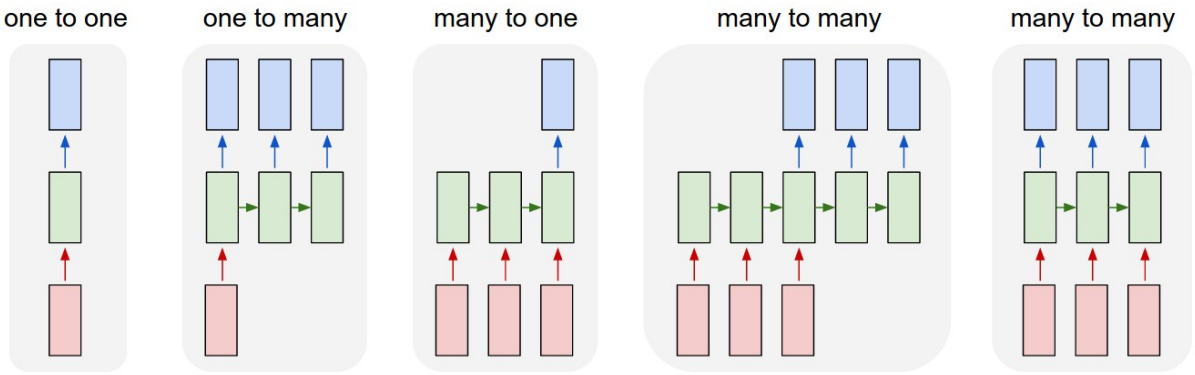
Figura 5 – Arquitetura Faster RCNN



Fonte: Retirado de Ren et al. (2017)

As RNN buscam resolver um problema das arquiteturas convencionais, chamadas de *feed foward* (ou ainda de *vanilla*) em que um exemplo é apresentado na primeira camada da rede e a entrada da próxima camada é o resultado da camada anterior, e assim sucessivamente, até obter uma resposta na última camada. Redes como a *Multilayer Perceptron* (MLP) e convolucionais são exemplos de redes *feed foward*. A limitação dessas redes é que não são capazes de lidar com dados sequencias ou quando a ordem dos dados é importante, como em uma frase, em que a alteração da ordem das palavras pode alterar o sentido.

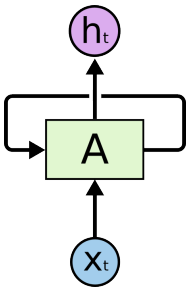
Figura 6 – Arquitetura *feed foward* e recorrentes



Fonte: Retirado de Karpathy (2015)

A Figura 6 apresenta uma simplificação das arquiteturas. A arquitetura *one to one* representa a rede *feed foward*, em que um exemplo é apresentado à rede que produz uma única

Figura 7 – Processo de recorrência



Fonte: Retirado de Karpathy (2015)

saída. As outras arquiteturas representam exemplos da RNN. A *one to many* demonstra uma arquitetura em que uma entrada produz uma sequência de saídas, como na legenda de imagens, em que uma imagem é dada como entrada na rede, que produz uma sequência de palavras. A *many to one* produz uma única saída a partir de várias entradas, como por exemplo uma sequência de palavras em que classifica-se um único sentimento (positivo ou neutro). E ainda existem duas variações da forma *many to many*, deslocada no tempo, como em uma tradução automática, em que é necessário ler uma sequência de palavras e depois gerar uma sequência de palavras como saída; e ainda a *many to many* sincronizada, que pode ser utilizada para realizar a classificação dos vários quadros de um vídeo.

O processo de recorrência ocorre conforme a Figura 7. Além de operar com a entrada e a matriz de pesos, agora também existe um estado interno h, análogo a uma memória interna. Assim o processamento das entradas também ocorre de forma diferente, em que o estado interno é dado pela Equação 2.1.

$$h_t = fw(h_{t-1}, x_t) \tag{2.1}$$

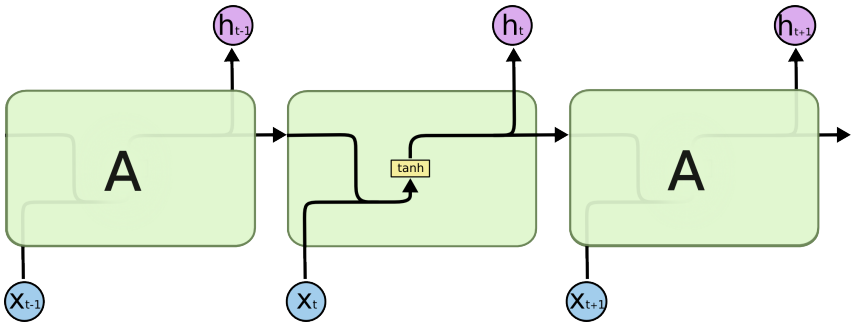
O tempo h_t é dado a partir de uma função com os pesos, fw, que recebe o tempo do estado anterior, h_{t-1}, juntamente com a entrada, x_t. Em outras palavras, ainda pode ser definida como definida na Equação 2.2.

$$h_t = tanh(W_{hh}h_{t-1} + W_{hx}x_t) \tag{2.2}$$

A matriz W_{hh} descreve a manipulação do estado interno no tempo h_{t-1} e W_{hx} atua sobre as entradas. A diferença para a rede *feed foward* é justamente a manipulação do estado interno $W_{hh}h_{t-1}$. A Figura 8 demonstra a execução da rede através dos vários tempos.

Na utilização da RNN para a geração de sentenças em língua natural, após o último tempo é executada uma função *softmax* para estimar a próxima palavra a partir do vocabulário, sendo que usualmente é utilizada a busca por feixe de tamanho 1, ou seja, é utilizada apenas a primeira palavra mais frequente.

Figura 8 – RNN em diferentes tempos



Fonte: Retirado de http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Assim como acontece nas RNA *feed foward*, o processo de aprendizado é realizado utilizando o *backpropagation*. Alguns dos problemas encontrados no treinamento da *feed foward* existem também na RNN. O problema do *vanish* no *exploding* do cálculo do gradiente quando existem muitas camadas na MLP acontece também na RNN quando existem muitos tempos para serem processados, dificultando lidar com dependências de longo prazo. Outro problema é a dificuldade de lidar com informações ruidosas ou pouco relevantes, visto que o estado interno sempre será atualizado.

Para resolver esses problemas foi criada a *Long Short Term Memory* (LSTM) por Hochreiter e Schmidhuber (1997), sendo um tipo de RNN. Para resolver o problema do *vanish* e *exploding* criaram um outro estado, chamado de *cell state*, além do estado interno, para ser utilizado como outra memória interna, sem nenhuma função, diferente da memória da RNN, fazendo com que o gradiente consiga fluir através dos vários tempos.

A LSTM também utiliza um conjunto de 3 *gates* que operam sobre o *cell state*. O primeiro *gate* é o *forget gate*, que é responsável por manter ou esquecer as informações. O *input gate* decide quais informações serão inseridas ou não, e o *output gate* que determina quais informações serão enviadas para a saída da rede ou para o próximo tempo.

Com essas alterações propostas pela LSTM é possível aumentar a dependência de longo prazo das RNN. A RNN possui uma dependência de cerca de 20 tempos, enquanto a LSTM aumenta para algumas centenas esse valor. Existem outras alterações que podem ser feitas para aumentar para até 1000 tempos.

2.2 Representação semântica com AMR

A motivação para utilizar uma representação semântica neste trabalho tem origem no trabalho de Karpathy e Fei-Fei (2015), no qual verificou-se a efetividade da utilização de relações sintáticas na tarefa de recuperação de informação, em contrapartida à abordagem tradicional de utilizar a sentença em língua natural. Em tal trabalho foi possível observar que o uso da sintaxe ajudou na recuperação da imagem a partir do texto e do texto a partir da imagem.

Partindo deste pressuposto de que as estruturas sintáticas beneficiaram a tarefa de recuperação de informação, neste trabalho persegue-se a hipótese de que a semântica pode, de modo semelhante, beneficiar a tarefa de geração de descrições para regiões de imagem.

Dentro da análise da língua natural, a semântica diz respeito ao significado (SPECIA; RINO, 2002), ao sentido da sentença, independente da forma como o texto está escrito. Assim, sentenças com estruturas lexicais e sintáticas diferentes, utilizando palavras e estruturas gramaticais distintas, podem possuir o mesmo significado. Como estabelecido por Jurafsky e Martin (2018), a semântica tem por finalidade resolver problemas que transcendem os outros níveis gramaticais (morfológico e sintático) e que precisam de algum processamento extra linguístico, um conhecimento de mundo.

A semântica pode ser representada de diversas maneiras. Segundo Jurafsky e Martin (2018), uma Representação Semântica (RS) deve ser capaz de expressar o significado de entradas diferentes (em relação a sua forma sintática) produzindo uma mesma forma de representação de significado (semântico). Ainda segundo Jurafsky e Martin (2018), a RS surge quando o processamento bruto das estruturas linguísticas, e suas derivações, são insuficientes para o processamento semântico. Para Abend e Rappoport (2017), uma RS é o reflexo de um determinado significado como entendido por um falante da língua.

A *Abstract Meaning Representation* (AMR) é uma RS que tem atraído grande interesse nos últimos anos. É uma RS simbólica para sentenças, que tem por objetivo unificar as tarefas semânticas, como aconteceu com as tarefas de sintaxe com a introdução dos bancos sintáticos (BANARESCU et al., 2013). O sucesso dos bancos sintáticos se deve ao fato de unificar as várias tarefas em um único processo, tornando necessário o uso de uma única ferramenta. Um exemplo de banco sintático clássico é o *Penn Treebank*[1].

Essa unificação de tarefas, no entanto, não acontecia com as RS. Eram necessários diferentes processos para cada uma das tarefas, como reconhecimento de entidades nomeadas, resolução de co-referências, anotação de relações semânticas, etc. (BANARESCU et al., 2013). A AMR tem como desafio unificar todas essas tarefas de maneira simples, legível do ponto de vista humano e facilmente gerada, da visão computacionalmente. A AMR suporta as relações de predicado (argumentos), incluindo papéis semânticos (adaptados do PropBank[2]), possibilitando uma grande quantidade de predicados, como verbos, co-referências, entidades nomeadas, etc. (ABEND; RAPPOPORT, 2017).

Atualmente a AMR suporta apenas semântica em nível de frase, sendo que foi criada com grande influência da língua inglesa, suportando assim a maioria dos fenômenos linguísticos do idioma inglês, sendo necessária a adaptação de alguns recursos para outras línguas. Trabalhos recentes em outras línguas tem obtido sucesso, como é o caso do português (ANCHIÊTA; PARDO, 2018), chinês (WANG et al., 2018), italiano, espanhol e alemão (DAMONTE; COHEN,

[1] https://web.archive.org/web/20131109202842/http://www.cis.upenn.edu/ treebank/
[2] Banco de preposições com seus respectivos argumentos.

2018).

Um aspecto importante de salientar é que a AMR não foi idealizada para ser uma representação de interlíngua, ou seja, não foi criada com a ideia de ser uma representação intermediária entre duas línguas.

A AMR faz extenso uso dos *framesets*[3] do PropBank, nomeando os verbos de acordo com seu sentido definido, algumas vezes até mesmo quando o verbo não aparece na frase. Por exemplo, em uma frase que possua o trecho *bond investor*, esse seria mapeado para RS de uma pessoa que realiza a ação de investir utilizando, assim, o verbo *invest*. O verbo *invest* será representado por seu sentido encontrado no PropBank, que no caso desse verbo existe apenas um único sentido, invest-01.

Uma RS AMR pode ser expressa de duas formas distintas: como um grafo ou como uma adaptação da anotação de PENMAN (MATTHIESSEN; BATEMAN, 1991). Na representação em formato de grafo, que possui uma única raiz, os conceitos são rotulados para os nós folhas e as relações entre os nós são representadas por meio das arestas. Quando é utilizada a anotação de PENMAN, os conceitos são atribuídos a uma variável, que por convenção é a simplificação da primeira letra, na forma de *(p / palavra)*, sendo a variável, *p*, uma instância de *palavra*. Isso é necessário para evitar a reescrita do conceito quando é necessário a sua utilização várias vezes.

Além das relações que os conceitos do PropBank dispõem, a AMR ainda possibilita a utilização de mais 100 outras relações para lidar com palavras-chave, como datas, unidades monetárias, relação de listas, etc. Uma especificação detalhada está descrita em suas diretrizes[4].

Para exemplificação das duas formas de representação serão usadas as seguintes frases retiradas de (ANCHIÊTA; PARDO, 2018):

The girl made adjustment to the machine.

The girl adjusted the machine.

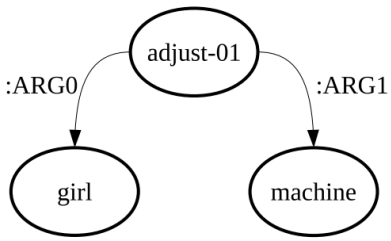
The machine was adjusted by the girl.

As frases possuem o mesmo sentido, de que a garota fez um ajuste na máquina, porém possuem estruturas diferentes. Na representação AMR, independente da forma, grafo ou notação de PENMAN, todas as frases terão uma única representação. As Figuras 9 e 10 ilustram essas duas formas de representação AMR.

O conceito principal identificado na frase foi *adjust*. No caso de existir mais de um verbo ou conceito, o principal conceito é rotulado como raiz. Em uma consulta ao PropBank, foi encontrado que o sentido do verbo *adjust* desempenha na frase é o seu primeiro e único

[3] Sentido da palavra, indicando uma estrutura sintática.
[4] https://www.isi.edu/ ulf/amr/help/amr-guidelines.pdf

Figura 9 – Notação de grafo



Fonte: Retirado de Anchiêta e Pardo (2018)

Figura 10 – Notação de PENMAN

```
(a / adjust-01
      :ARG0 (g / girl)
      :ARG1 (m / machine))
```

Fonte: Retirado de Anchiêta e Pardo (2018)

sentido[5], que diz respeito a uma pequena mudança. Os argumentos para o sentido de *adjust-01* são apresentados a seguir.

ARG-0: Causador do ajuste.

ARG-1: Coisa ajustada.

ARG-2: Finalidade para qual ARG-1 foi ajustada.

ARG-3: Estado inicial antes do ajuste.

ARG-4: Razão do ajuste.

Na frase, o causador do ajuste é *girl* e, portanto, ele é anotado como o primeiro argumento de *ajust* (ARG-0). O objeto que sofreu a ação do verbo, causada por ARG-0, é *machine*, completando todas as informações da sentença. Apesar do sentido do verbo possuir mais argumentos, não é obrigatório preencher todos caso não exista informação suficiente na frase.

Um aspecto importante de destacar é que o formalismo AMR realiza algumas normalizações nas estruturas da sentença. Para o exemplo apresentado anteriormente, foi perdida a informação do tempo verbal em que aconteceu a ação relatada na frase, ou seja, não se sabe se a máquina foi ajustada, se está sendo ajustada ou se ainda será ajustada. Essas informações descartadas são conhecidas como açúcares sintáticos.

Do ponto de vista computacional, a AMR geralmente é representada na forma linear a partir da notação de PENMAN. Para o exemplo ilustrado na Figura 10, a representação AMR linearizada seria: *(a / adjust-01 :ARG0 (g / girl) :ARG1 (m / machine))*.

Para a geração automática das representações AMR, utiliza-se um *parser*. Assim, o *parser* é a ferramenta responsável por transformar uma sentença em língua natural em sua correspondente representação AMR, ou fazer o processo inverso de transformar uma representação AMR na sentença língua natural correspondente. Para diferenciar esses dois processos, geralmente utiliza-se o nome de *Text-to-AMR* para a geração de AMR a partir de uma sentença

[5] http://verbs.colorado.edu/propbank/framesets-english-aliases/adjust.html

em língua natural e *AMR-to-Text* para a transformação de AMR para uma sentença em língua natural.

Capítulo 3

GERAÇÃO DE LÍNGUA NATURAL

O Processamento (Automático) de Língua Natural (PLN), segundo Chowdhury (2003), é a área de pesquisa e aplicação que explora como os computadores podem ser utilizados para compreender e manipular textos em língua natural ou fala para realizar tarefas úteis. Dentro do PLN, um dos grandes desafios é fazer a **geração automática da língua natural** como produto direto ou indireto de diversas aplicações.

Geração de Língua Natural (GLN) começou a ganhar atenção na segunda metade do século 20, no contexto da tarefa de tradução automática (GARCÍA-MÉNDEZ et al., 2019). Todavia, foi a partir da décadas de 1970 e 1980 que a GLN começou a ganhar interesses de pesquisas.

Segundo Reiter e Dale (1997), GLN é o campo que estuda formas de gerar informação compreensível em alguma língua natural humana a partir de uma representação não linguística. GLN combina conhecimento linguístico com o domínio da aplicação para produzir, de forma automática, documentos, relatórios, mensagens de ajuda e outros tipos de textos (REITER; DALE, 1997).

O uso mais comum de tecnologias de GLN está na criação de aplicações capazes de apresentar informações para as pessoas de modo que possam ser facilmente compreendidas (REITER; DALE, 1997). Diversas aplicações fazem uso de informações técnicas que são entendidas apenas por especialistas na área, como sistemas de vôos, manipulação de ações na bolsa de valores entre outros (REITER; DALE, 2000). Em diversos casos, é necessário exibir essas informações para pessoas que não possuem o mesmo conhecimento que os especialistas da área, sendo necessária a transformação da informação original em uma versão compreensível por não especialistas (REITER; DALE, 1997).

Contudo, essas definições relatam apenas uma faceta de GLN, onde os primeiros sistemas recebiam como entrada palavras, frases e até textos completos para produzir um novo texto como saída (GARCÍA-MÉNDEZ et al., 2019). Com as grandes mudanças sofridas pela área nos últimos anos, tem se tornado difícil definir o que é GLN (REITER; DALE, 1997).

Uma definição mais abrangente e atual seria considerar GLN como a área que estuda

a produção e criação de novas técnicas e abordagens para auxiliar na geração ou produzir (diretamente ou indiretamente) uma representação em língua natural, a partir de uma outra representação (de língua natural ou não).

Aplicações de GLN tem conquistado grande interesse nos últimos anos devido, em parte, à grande quantidade de informações existentes na internet que de alguma forma precisam ser processadas e analisadas para, por exemplo, extrair algum conhecimento. A seguir, são listadas algumas aplicações que fazem uso, direta ou indiretamente, da GLN (GATT; KRAHMER, 2018).

- Tradução automática de textos (WU et al., 2016; CASELI et al., 2006);
- Sumarização automática de textos (PARDO et al., 2003);
- Simplificação de textos (ALUÍSIO et al., 2008; SIDDHARTHAN, 2014; MACDONALD; SIDDHARTHAN, 2016);
- Geração automática de paráfrases (KAUCHAK; BARZILAY, 2006);
- Geração automática de perguntas para textos e conjunto de sentenças (BROWN et al., 2005; RUS et al., 2010).

Existem aplicações ainda que geram textos a partir de informações, os quais são conhecidas como dado-texto. Como é o caso de um robô-jornalista do *New York Times* que 3 minutos após um terremoto, ocorrido em *Beverly Hills* no estado da Califórnia nos Estados Unidos, em 17 de março de 2014, foi capaz de criar uma reportagem fornecendo detalhes como magnitude e localização do terremoto (GATT; KRAHMER, 2018).

Outra aplicação bastante difundida e que tem conquistado bons resultados é a geração de textos a partir de dados meteorológicos como em Reiter et al. (2005), no qual um texto com a previsão do tempo é gerado a partir dos dados das informações meteorológicas.

Dentre as diversas aplicações que se beneficiam da GLN, de especial interesse para este trabalho são as duas descritas nas seções a seguir: geração de legenda de imagem (Seção 3.1) e geração de descrições para regiões de imagem (Seção 3.2).

3.1 Geração de legenda de imagem

A tarefa de Geração de Legenda de Imagem (GLI), do inglês *image captioning*, pode ser entendida como a área de intersecção entre visão computacional e PLN em que busca-se gerar descrições em língua natural a partir de imagens (YAGCIOGLU et al., 2015). É o processo automático de prover uma legenda, em língua natural, para uma imagem, o que pode ser feito por meio da rotulação de palavras-chaves relevantes relacionadas ao conteúdo da imagem (SRIVASTAVA; SRIVASTAVA, 2018). Segundo Sumathi e Hemalatha (2011), essa rotulação é o

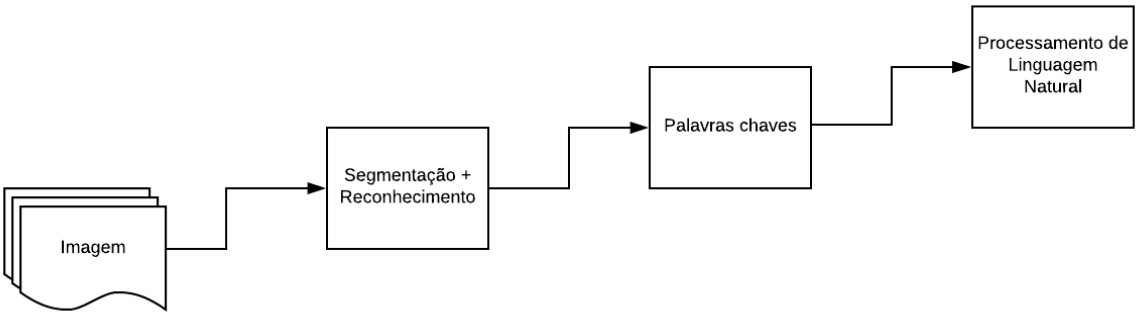
processo em que um sistema computacional atribui automaticamente metadados na forma de legendas ou palavras-chave a uma imagem digital.

Para Wang et al. (2016), a geração de legendas de imagens tem a capacidade de reconhecer objetos visuais em imagens, as interações semânticas entre esses objetos, bem como a interação visual-linguística, além de "traduzir" a compreensão (visual) para a descrição em forma de sentenças em língua natural.

A tarefa tem atraído o interesse tanto de profissionais da área de visão computacional, para melhor representar os conceitos e relações de uma imagem, como de PLN, para tentar expressar, em palavras, a informação presente na imagem (SRIVASTAVA; SRIVASTAVA, 2018).

A Figura 11 traz uma visão geral sobre o processo de geração de legenda de imagens. A imagem completa é dada como entrada para a etapa de segmentação e reconhecimento dos objetos presentes. A partir de então é feita a extração das palavras-chave que servirão de insumo para a etapa de PLN que irá gerar a legenda. Alguns autores incluem mais algumas etapas neste processo, como o recurso de atenção e modelagem de contexto (SRIVASTAVA; SRIVASTAVA, 2018).

Figura 11 – Fluxo do processo de geração de legenda de imagem



Fonte: Adaptado de Srivastava e Srivastava (2018)

Na tarefa de geração de legenda de imagem, nem todos os elementos da imagem possuirão correspondência na legenda em língua natural. Geralmente, apenas o(s) principal(is) elemento(s) detectados na imagem estará(ão) mencionado(s) na legenda. Os outros elementos são importantes para modelar o contexto da imagem e dar suporte à legenda como um todo. Essa análise de cena é extremamente importante e possui grande influência no resultado final da legenda.

A tarefa de geração de legendas para imagens é relacionada ao objeto de estudo deste trabalho: a geração de descrições para regiões de imagem. Por isso, no Capítulo 4 serão apresentados trabalhos relacionados a este como forma de descrever as principais estratégias utilizadas na atualidade. Contudo, ressalta-se que o foco deste trabalho é a geração de descrições para regiões de imagem.

3.2 Geração de descrições para regiões de imagem

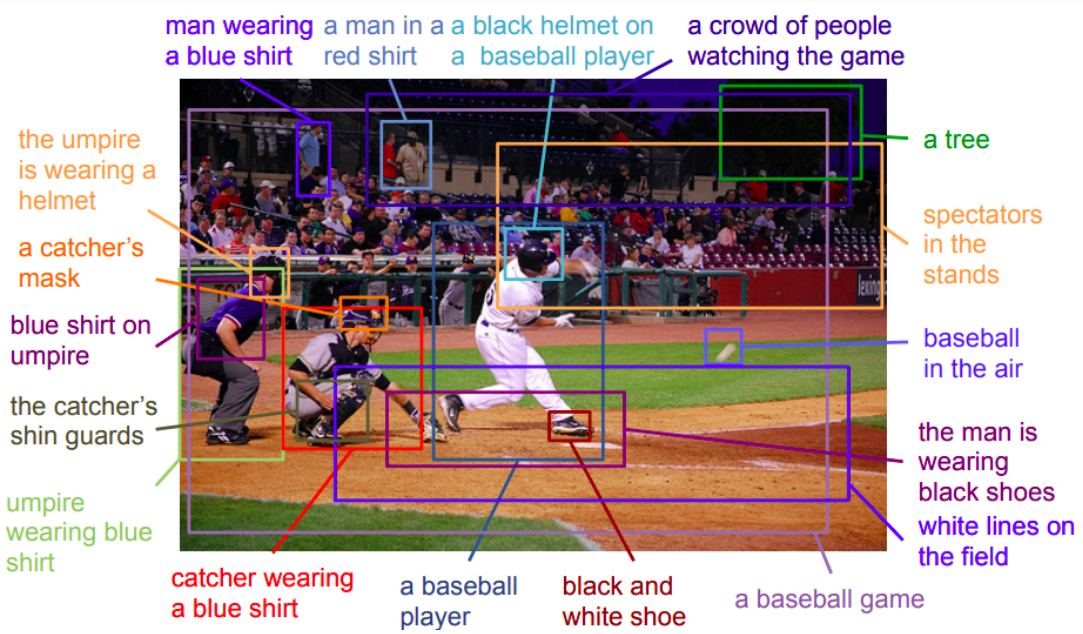
Enquanto a GLI tem por objetivo produzir uma sentença que descreva a imagem por completo, a **Geração de Descrições para Regiões de Imagem** (GDRI) tem como finalidade prover uma maior quantidade de informações para uma mesma imagem. Uma imagem geralmente possui uma grande quantidade de informações de vários aspectos. Descrever todas essas informações em LN em uma única frase pode ocasionar a perda de informações, seja por limitações de tecnologia, limitação do tamanho da sentença ou até mesmo da língua escolhida para descrição.

Comparando com tarefas similares e multidisciplinares, como classificação de objetos, reconhecimento de objetos e até mesmo a GLI, a GDRI é uma área emergente e consideravelmente mais nova que as demais (ZHANG et al., 2015). Enquanto a GLI produz uma sentença para a imagem completa, a GDRI deve produzir uma sentença (ou frase) para cada uma das várias regiões existentes na imagem.

Cada imagem possui várias regiões, para as quais a GDRI deve gerar suas respectivas descrições em língua natural. De modo similar ao que ocorre para a GLI, a sentença produzida pela GDRI deve expressar a maneira que os objetos relacionam-se entre si e seus atributos (ZHANG et al., 2015). A Figura 12 ilustra a GDRI para uma imagem.

Como algumas das aplicações da GDRI, cita-se a possibilidade de auxilio às pessoas com deficiência visual ou com baixa visão, de modo que uma informação equivalente à imagem seja exibida em LN, ou ainda a recuperação de imagens em um sistema de busca que tenha como entrada sentenças em LN (ZHANG et al., 2015).

Figura 12 – Descrições de regiões de imagem



Fonte: Retirado de Yang et al. (2017)

Uma complexidade maior que a GDRI possui em relação à GLI é que uma legenda de imagem, de modo geral, descreve as principais informações da imagem, sem a necessidade de descrever (quase) todos os elementos. Em oposição, a GDRI descreve muito mais informações por imagem, em virtude de uma imagem possuir várias regiões. Algumas dessas descrições podem conter elementos considerados irrelevantes para a GLI.

Os trabalhos recentes fazem uso extensivo de técnicas de RNA para a GDRI e tarefas similares. Tais técnicas podem ser divididas em duas partes: a detecção e extração das características das regiões da imagem; e a geração da sentença de fato. A primeira parte, responsável pela imagem, é realizada através da RNA do tipo CNN, enquanto a segunda parte, encarregada pela geração da sentença, utiliza uma RNA LSTM.

Em relação à geração da sentença em língua natural que descreve uma região da imagem, os trabalhos atuais não utilizam nenhuma representação semântica intermediária e, portanto, são passíveis de serem afetados com problemas intrínsecos da LN, tal como a ambiguidade. Com o objetivo de verificar se representações semânticas podem beneficiar a GDRI, neste trabalho, os modelos foram treinados com representações semânticas AMR, sendo essas posteriormente transformadas em LN.

3.3 Aplicações utilizando AMR

Conforme mencionado na Seção 2.2, a AMR é utilizada como forma de representação semântica, de modo que sentenças com estruturas diferentes, do ponto de vista morfológico e sintático, mas que possuem o mesmo significado, sejam representadas de uma única forma. Para algumas aplicações nas quais a AMR é empregada, ela pode ser vista como uma representação intermediária para uma posterior transformação para a LN, por ser uma forma mais controlada e simplificada de uma estrutura que pode ser transformada em LN.

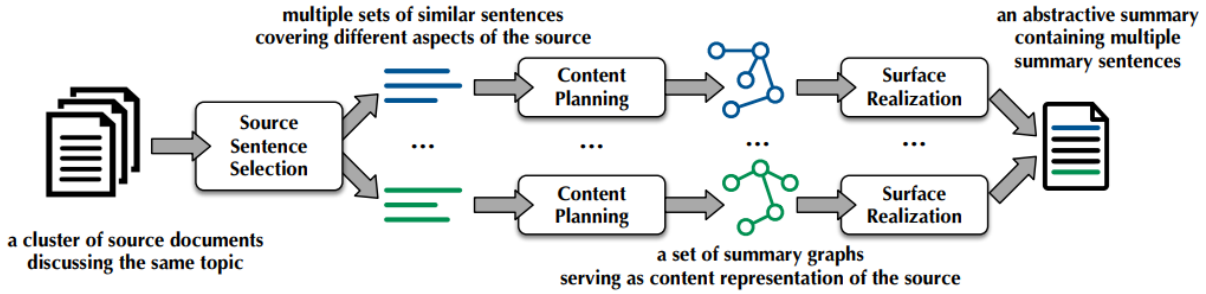
As aplicações que transformam LN em AMR são conhecidas como *parsers* ou conversores *Text-to-AMR*, como é o caso do trabalho de Lyu e Titov (2018). O processo contrário, que transforma AMR em LN é chamado de *AMR-to-Text*, como o trabalho de Song et al. (2018).

Dentre as aplicações de PLN, a tradução automática (TAMCHYNA et al., 2015) e a sumarização automática (LIU et al., 2015; LIAO et al., 2018) são as mais populares, que evidenciam os benefícios do uso de AMR.

O trabalho de Liao et al. (2018), por exemplo, teve por objetivo realizar a sumarização automática multidocumento. A abordagem, ilustrada na Figura 13, é composta por três etapas: (1) seleção das sentenças de origem a partir do conjunto de artigos de notícias, no qual a partir desse conjunto ocorre a seleção de frases semelhantes que abrangem diferentes aspectos do mesmo tópico; (2) planejamento de conteúdo, em que é produzido um grafo AMR a partir do conjunto de sentenças da etapa anterior; (3) geração superficial da sentença em língua natural a

partir do grafo do sumário gerado na etapa anterior.

Figura 13 – *Pipeline* da arquitetura de Liao et al. (2018)



Fonte: Retirado de Liao et al. (2018)

A seleção de sentenças de origem ocorre agrupando documentos com o mesmo tópico, no qual cada sentença deve cobrir um aspecto diferente do tópico. Para isso, é utilizado um cálculo de similaridade entre pares de sentenças. Para o planejamento de conteúdo, a partir das sentenças selecionadas e suas respectivas representações AMR, ocorre a consolidação dos grafos por meio da fusão dos nós que representam o mesmo conceito (com base no casamento da forma superficial). A geração superficial da sentença em língua natural ocorre convertendo os grafos AMR resultantes da fusão para a notação de PENMAN e, posteriormente, convertendo-a para língua natural através de um *parser AMR-to-Text*.

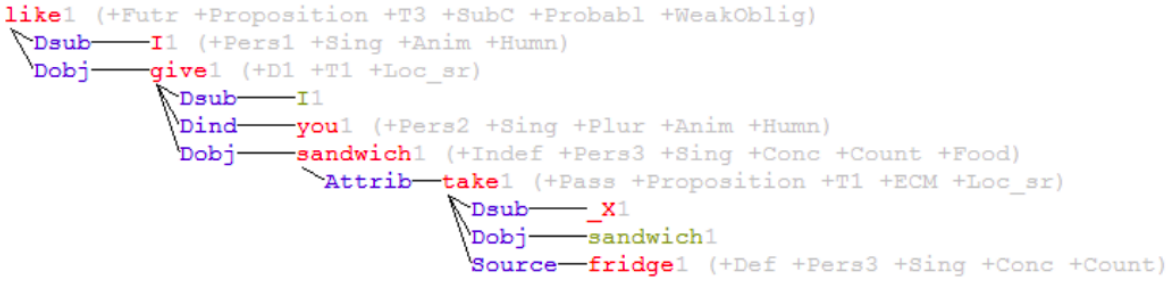
Como medida de avaliação, os autores usaram a *Recall-Oriented Understudy for Gisting* (ROUGE) (LIN, 2004) para comparar os modelos, com unigramas (ROUGE-1), bigramas (ROUGE-2) e *skip gram* (de 1 a 4 – ROUGE-SU4), entre os sumários produzidos pelo modelo com AMR e os *baseline*, em comparação com os de referência. Os resultados obtidos pelo modelo com AMR foram superiores na avaliação com o *dataset* da *Text Analysis Conference*[1] (TAC) de 2011: 33,1 contra 44,1 na ROUGE-1; 7,5 contra 8,3 na ROUGE-2; e 11,1 contra 13,5 na ROUGE-SU4.

Outro trabalho que também utilizou AMR e os resultados obtidos foram melhores que os trabalhos anteriores foi o de Tamchyna et al. (2015), no qual uma RS foi usada na tarefa de tradução automática como uma interlíngua.

O formato em que a RS é empregada é um pouco diferente da ideia inicial proposta por Banarescu et al. (2013). É utilizada uma variação da lógica formal em que a AMR também pode ser descrita, criada por Vanderwende et al. (2015). Essa variação é utilizada no formato de árvores direcionadas rotuladas, cujos nós correspondem às palavras da sentença. As arestas definem a relação semântica entre os nós. Informações linguísticas adicionais, como o tempo verbal, sub-categorização, gênero, número, entre outros, também são representados nos nós. Um é ilustrado na Figura 14 para a frase "*I would like to give you a sandwich taken from the fridge*".

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

Figura 14 – Variação da Lógica Formal



Fonte: Retirado de Tamchyna et al. (2015)

As sentenças em inglês são convertidas para esse formato de lógica formal através do trabalho de Vanderwende et al. (2015) para depois serem utilizadas pelo modelo proposto que as transforma em sentenças em francês. Como *baseline*, foi empregada uma variação do modelo oculto de Markov. Para o modelo, foi usado o método de entropia máxima para modelar a probabilidade condicional, pois assim foi possível utilizar um conjunto de recursos comumente utilizados nos modelos de tradução (normalmente informações que podem ser extraídas através de algum *part of speech tagger*).

Os resultados obtidos demonstraram uma ligeira melhora do modelo proposto (17,55) contra o *baseline* (17,41) na avaliação da BLEU (valor de n-grama não informado) utilizando o *dataset Workshop on Statistical Machine Translation* (BOJAR et al., 2013) de 2013 (ano mais recente reportado).

3.4 Medidas de avaliação

Como os modelos que geram língua natural tentam, de alguma forma, automatizar alguma tarefa humana em seu último estágio, em cenários ideais, essa avaliação deveria ocorrer de forma manual, ou seja, pessoas avaliando a qualidade das informações geradas pelos modelos. No entanto, quando a quantidade de informação a ser avaliada é muito grande, essa avaliação manual pode demandar várias pessoas durante um longo período de tempo (semanas e até mesmo meses). Como essa avaliação demanda tempo e recurso humano, não é possível avaliar os modelos após pequenas alterações. Dessa forma, frequentemente é necessário automatizar esse processo.

Como alternativa para esse processo manual, foram criadas medidas que procuram reproduzir a avaliação humana de forma automática. A seguir, são apresentadas algumas medidas automáticas de avaliação mais utilizadas no contexto GLN, divididas entre aquelas projetadas para avaliar a língua natural – BLEU, METEOR – e outras que têm foco na avaliação de conteúdo, sendo uma especificamente usada para GDRI – mAP – e outra específica para AMR – SMATCH.

3.4.1 Medidas que avaliam a língua natural

Esta seção descreve as medidas utilizadas neste trabalho para avaliar as sentenças em forma de LN. De modo geral, tais medidas procuram comparar o grau de semelhança entre uma sentença gerada automaticamente (candidata) e outra de referência (considerada correta), porém por meio de estratégias diferentes. Os seus valores são representados de 0 a 1, em que quanto mais próximo de 1, maior a semelhança entre as sentenças.

3.4.1.1 *Bilingual Evaluation Understudy*

A *Bilingual Evaluation Understudy* (BLEU) é uma medida de avaliação criada por Papineni et al. (2002) com o objetivo de avaliar, de forma automática, a tradução automática e que, recentemente, também passou a ser utilizada para avaliar modelos de GLN. A BLEU é definida formalmente como apresentado na Equação 3.1.

A BLEU é calculada a partir dos n-gramas, que são sequências de n palavras (ou *tokens*). Dessa forma, unigrama é uma única palavra, bigrama duas palavras consecutivas, e assim sucessivamente. Para exemplificar, suponha as seguintes sentenças de referência e candidatas geradas por um modelo.

Referência 1: *The cat is on the mat*

Referência 2: *There is a cat on the mat*

Candidata: *The cat the cat on the mat*

As referências são as sentenças corretas que o modelo deveria gerar, enquanto a candidata é a gerada pelo modelo. Suponha que a avaliação ocorra com bigramas, dessa forma a sentença candidata forma os seguintes bigramas não repetidos: *the cat*; *cat the*; *cat on*; *on the*; e *the mat*.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count(n\text{-}gram)} \tag{3.1}$$

Na Equação 3.1, $count_{clip}$ é o maior número de vezes que o bigrama da sentença candidata aparece em uma das referências. No exemplo apresentado anteriormente, os valores seriam: 0 para *cat the* e 1 para *the cat*, *cat on*, *on the* e *the mat*, sendo que esse último bigrama aparece nas duas referências, mas apenas uma única vez em cada. Assim, o valor do somatório do numerador é 4. Já o $count$ é calculado de acordo com o número de vezes que os n-gramas acontecem na sentença candidata. Na sentença de exemplo, esses valores são: 2 vezes para *the cat* e 1 vez para *cat the*, *cat on*, *on the* e *the mat*, resultando em 6 para o somatório no denominador.

Em casos em que a sentença candidata é curta, as palavras presentes são extremamente prováveis de ocorrem na sentença de referência. Dessa forma é necessário penalizar quando sentenças candidatas forem pequenas, somando com todos os n-gramas, na forma:

$$BLEU = BP * exp\left(\frac{1}{N}\sum_{n=1}^{N} p_n\right) \tag{3.2}$$

em que *Brevity Penalty* (BP) é:

$$BP = \begin{cases} 1,\ se\ |candidata|\ >\ |referencia| \\ caso\ contrario,\ exp(1\ -\ |candidata|\ /\ |referencia|) \end{cases}$$

Quando a BLEU é reportada, é necessário indicar o tamanho de n-gramas considerado em seu cálculo. Dessa forma, a BLEU é apresentada na forma BLEU-n, onde n indica o tamanho do n-grama. Essa convenção será utilizada nos capítulos e seções seguintes para reportar os valores obtidos nas avaliações.

Uma desvantagem que a BLEU possui é que dependendo do valor de n-grama, o cálculo pode não refletir a real qualidade da sentença. Por exemplo, utilizando como BLEU-1, se a candidata possuir as mesmas palavras da referência, mas em uma ordem totalmente diferente, será obtido um valor alto, mas que não reflete, necessariamente, a qualidade da sentença candidata.

De modo geral, a BLEU exige que exatamente as mesmas palavras sejam usadas, na mesma ordem, na referência e na candidata. Para algumas tarefas, isso pode ser observado como uma qualidade se assim exigir. No entanto, a BLEU penaliza tarefas em que é possível utilizar palavras diferentes, mas com a frase mantendo o mesmo significado, como ocorre na GDRI.

Dessa forma, a BLEU não é uma medida adequada para a GRDI, visto que a não ocorrência exata das mesmas palavras da referência na candidata não indica necessariamente que a sentença predita pelo modelo (candidata) seja ruim. Contudo, uma vez que valores de BLEU são reportados em alguns trabalhos relacionados, a BLEU é uma das medidas utilizadas neste trabalho, mas dando-se ênfase apenas para o valor de BLEU-1 para verificar se as sentenças preditas possuem as mesmas palavras da referência.

3.4.1.2 *Metric for Evaluation of Translation with Explicit Ordering*

Quando uma aplicação é avaliada de acordo com a medida BLEU, é necessário que a saída seja igual a referência, ou seja, que utilize exatamente as mesmas palavras para que possua uma alta pontuação. Outra medida de avaliação também proposta para avaliar sistemas de tradução automática, a *Metric for Evaluation of Translation with Explicit Ordering* (METEOR) de Banerjee e Lavie (2005), busca solucionar esse problema. Para tanto, ela considera não apenas a correspondência entre palavras que são idênticas entre a candidata e a referência, mas também mede a correspondência de palavras que são simples variantes morfológicas ou sinônimos.

Para isso, o alinhamento (conjunto de mapeamento entre unigramas) acontece entre as sentenças (referência e candidata) através de diferentes módulos e critérios. O primeiro módulo é chamado de exato, no qual dois unigramas são alinhados se forem exatamente os mesmos. Por exemplo, o unigrama "computador" na referência é alinhado com "computador" na candidata, mas não com "computadores". O segundo módulo é o de estemização, no qual dois unigramas são alinhados se eles tiverem o mesmo radical após utilizar o Porter Stemming[2]. Assim, nesse módulo, a palavra "computador" é alinhada com "computadores", pois possuem o mesmo radical ("comput"). O terceiro módulo é o de sinônimo, no qual unigramas entre as sentenças candidata e de referência são alinhados se tiverem o mesmo significado após consulta na WordNet[3].

Esses são os módulos padrão na METEOR. No entanto, é possível adicionar outros módulos conforme a necessidade e assim estender e ajustar a métrica de acordo com a aplicação pretendida. O alinhamento de cada unigrama candidato deve ocorrer entre 0 e no máximo 1 unigrama na referência. Quando existir mais de um alinhamento possível, o alinhamento selecionado é o que possuir o menor número de cruzamentos entre os mapeamentos.

A precisão é calculada como:

$$P = \frac{m}{w_t} \tag{3.3}$$

na qual m é o número de unigramas da sentença candidata alinhados com a referência e w_t, o total de unigramas na sentença candidata. O revocação é definida como:

$$R = \frac{m}{w_r} \tag{3.4}$$

em que w_r é o total de unigramas da sentença de referência. A precisão e a revocação são combinadas na $Fmean$ utilizando a média harmônica, como :

$$Fmean = \frac{10PR}{R + 9P} \tag{3.5}$$

Essas medidas calculam a relação de palavras isoladas, mas não em relação a segmentos maiores, tanto na referência quanto na candidata. Para calcular esses segmentos maiores, os unigramas da referência são agrupados no menor número possível de *chuncks* (conjunto de unigramas que são adjacentes na referência e candidata). Quanto maior o mapeamento adjacente, menor o número de *chuncks*. A penalidade é calculada como:

$$penalty = 0.5 * \left(\frac{c}{u_m}\right)^3 \tag{3.6}$$

[2] https://tartarus.org/martin/PorterStemmer/
[3] https://wordnet.princeton.edu/

na qual c é o número de *chunks* e u_m é o número de unigramas mapeados. Finalmente, o *score* é calculado como:

$$score = Fmean * (1 - penalty). \tag{3.7}$$

Para calcular a pontuação entre mais de uma sentença como referência é necessário obter o *score* para cada referência com a candidata e então selecionar a que possuir maior valor.

A vantagem na utilização da METEOR em relação a BLEU é que a mesma permite uma maior variação das palavras, pois além do módulo de alinhamento exato (assim como na BLEU), também faz uso dos módulos de estemização e de sinônimos.

Como principal desvantagem de utilizar a METEOR para a GDRI é que, assim como acontece com a BLEU, se a referência e a candidata possuírem informações diferentes sobre a *bbox*, sendo que ambas as sentenças estão corretas, mas descrevem elementos diferentes, a medida não irá refletir a real qualidade da sentença predita pelo modelo.

A partir disso, ressalta-se também que a METEOR foi utilizada exclusivamente para verificar o número de palavras da sentença predita pelo modelo em relação a referência, sendo a diferença para a BLEU a utilização de seus módulos de estemização e de sinônimos.

3.4.2 Medidas que avaliam conteúdo

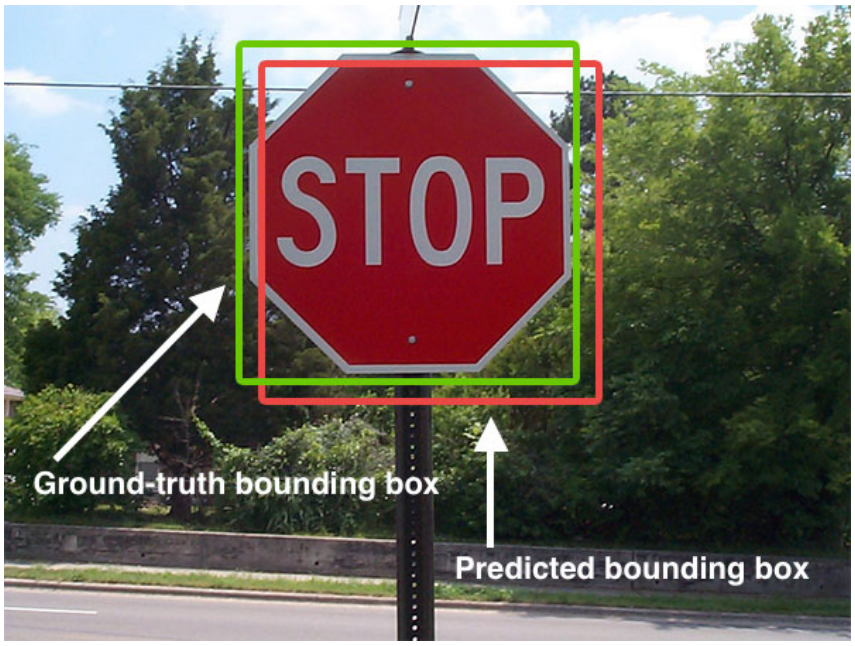
Esta seção descreve as medidas utilizadas neste trabalho para avaliação das descrições geradas dando ênfase para o conteúdo que carregam e não para as palavras e n-gramas que contém. Para tanto, foram utilizadas duas medidas: a *Mean Average Precision* (mAP), comumente usada nos trabalhos de GDRI, e a *Semantic Match* (SMATCH), proposta para avaliar grafos AMR. A mAP é calculada comparando-se a descrição candidata com a descrição e a *bbox* de referência. Seus valores começam em 0 mas não têm um limite superior, sendo que quanto maior, melhor é a qualidade da descrição candidata gerada. Já a SMATCH é calculada comparando-se o grafo AMR candidato com o grafo AMR de referência. Seus valores variam entre 0 e 1 e quanto mais próximo de 1, maior é a semelhança entre os grafos AMR sendo comparados.

3.4.2.1 *Mean Average Precision*

Uma métrica mais apropriada para avaliação da GDRI é a *Mean Average Precision* (mAP). A mAP foi criada por Everingham et al. (2009) como forma de avaliar sistemas de detecção de objetos. A detecção de objetos é a localização de um objeto, através da delimitação de sua *bbox*, e a classificação do mesmo. Ou seja, o modelo deve predizer a região da *bbox* e ser capaz de dizer qual objeto está dentro dela.

A mAP foi adaptada para ser utilizada em outras tarefas como a recuperação de informação e a avaliação de modelos de GDRI. Em virtude disso os trabalhos relacionados reportam

Figura 15 – Exemplo de *Intersection over union*



Fonte: Retirado de Rosebrock (2020)

valores da mAP no Capítulo 4.

Primeiramente, para calcular a mAP é preciso obter os valores da precisão e revocação calculados conforme apresentado nas Equações 3.9 e 3.10. Para ilustrar esse cálculo, considere o exemplo da Figura 15. Para medir o quão correta está a *bbox*, é necessário primeiro calcular a intersecção sobre a união (*Intersection over Union* – IoU) da *bbox* de referência (*ground truth*) e a predita (*predicted*). A IoU é definida pela Equação 3.8, que nada mais é do que a área de intersecção entre as *bbox* sobre a área de união das mesmas.

$$IoU = \frac{Intersection}{Union} \tag{3.8}$$

A partir disso, é possível calcular a precisão e a revocação. Para isso, considera-se como Verdadeiros Positivos (VP) as *bbox* em que a IoU é maior que um limiar (o mais comum é 0,5) e o objeto foi corretamente classificado. Caso a IoU é menor que o limiar são considerados Falsos Positivos (FP). A precisão é dada pela Equação 3.9.

$$Precision = \frac{VP}{VP + FP} \tag{3.9}$$

Para calcular a revocação, é necessário obter os valores "negativos". No entanto, os Verdadeiros Negativos não precisam ser calculados, pois são consideradas áreas em que não existe uma *bbox* de referência e o modelo também não predisse uma *bbox*. Dessa forma, os Falsos Negativos (FN) são as áreas em que a *bbox* é maior que o limiar estabelecido, mas a classificação está incorreta; ou áreas em que existe uma *bbox* de referência, mas o modelo não foi capaz de predizer a região. A revocação é obtida através da Equação 3.10.

$$Recall = \frac{VP}{VP + FN} \tag{3.10}$$

A *Average Precision* (AP) pode ser definida, de modo geral, como a área sob a curva de um gráfico precisão-revocação. A revocação é definida como a proporção de todos os exemplos positivos classificados acima de um limiar, e a precisão é a proporção de todos os exemplos acima dessa classificação que são da classe positiva. A AP resume a forma da curva precisão/revocação e é definida em um conjunto de onze níveis de interpolação igualmente espaçados entre 0 e 1 (0, 0,1, ..., 1) (Equação 3.11).

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{interp(r)} \tag{3.11}$$

na qual a precisão para cada nível de revocação r é interpolada tomando a precisão máxima medida para um método para o qual a revocação correspondente excede r, conforme Equação 3.12, onde $p(\widetilde{r})$ é a precisão para a revocação $\widetilde{r}$.

$$p_{interp}(r) = \max_{\widetilde{r}:\widetilde{r} \geq r} p(\widetilde{r}) \tag{3.12}$$

A mAP neste caso é definida como a média da AP para vários valores de IoU e para todas as categorias de objetos do sistema de detecção, daí origina-se o seu nome. Para a sua aplicação na tarefa de avaliação de modelos de GDRI, é necessário realizar algumas adaptações. Dessa forma, a seguir será descrito como a avaliação ocorreu, seguindo o trabalho e código fonte disponibilizado por Johnson et al. (2016).

Primeiramente, é necessário definir os VP e os FP. Para isso, dois limiares são necessários: a IoU e a pontuação mínima das descrições. A IoU é utilizada da mesma forma que na AP para

a detecção de objetos, conforme Equação 3.8, no qual deve ser maior ou igual aos seguintes limiares: 0,3, 0,4, 0,5, 0,6 e 0,7.

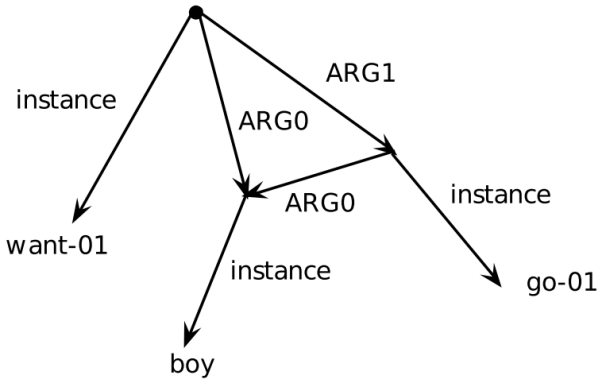
A pontuação mínima entre a sentença de referência e a predita é calculada utilizando a medida METEOR, sendo que os valores devem ser maiores que os limiares estipulados (0, 0,05, 0,1, 0,15, 0,2, 0,25). Os VP e os FP são obtidos verificando se os exemplos possuem valores maiores de pontuação mínima e IoU para cada combinação possível entre os limiares de pontuação mínima e IoU.

A precisão e a revocação são calculados da mesma forma que definidos na AP para cada combinação de pontuação mínima e IoU, conforme Equações 3.9 e 3.10, respectivamente. Diferente da AP para avaliação de sistemas de reconhecimento de objetos, na avaliação para GRDI são utilizados 100 valores de interpolação ($r = \{0, 0, 01, 0, 02..., 0.99\}$), multiplicando os valores de revocação que sejam maior que a interpolação corrente (r) pela precisão. Posteriormente, é calculada a média aritmética dos valores de todas as combinações, obtendo assim um valor único referente a avaliação.

3.4.2.2 *Semantic Match*

As medidas de avaliação automática descritas até então são amplamente utilizadas para a avaliação de métodos de GLN e recuperação de informação. Contudo, como este trabalho utiliza o formalismo de representação semântica AMR, uma medida aplicada especificamente para avaliação de grafos AMR também foi utilizada: a *Semantic Match* (Smatch) de Cai e Knight (2013). Para ilustrar o funcionamento da Smatch, considere o exemplo apresentado na Figura 16 para a frase "*the boy wants to go*".

Figura 16 – Alternativa a notação de grafo



Fonte: Retirado de Cai e Knight (2013)

A diferença para a notação usual de grafo, apresentada na Figura 9, é que os vértices não são rotulados. Neste caso, os rótulos são representados como descendentes, com a aresta indicada com *instance*, ou seja, que o vértice é uma instância do seu descendente indicado por

instance. De maneira análoga, também é possível representar em formato de lógica proposicional, conforme apresentado a seguir.

$\exists$ *A, B, C*
instance (A, want-01) $\wedge$
instance (B, boy) $\wedge$
instance (C, go-01) $\wedge$
ARG0 (A, B) $\wedge$
ARG1 (A, C) $\wedge$
ARG0 (C, B)

A representação AMR usando a lógica pode ser expressa de duas formas de triplas, sendo: (1) *relação (variável, conceito)*, como nas três primeiras triplas do exemplo anterior, ou (2) *relação (variável, variável)*, como nas últimas três triplas do exemplo anterior.

A Smatch tem como objetivo verificar a quantidade de triplas iguais. Para ilustrar seu cálculo, suponha a sentença "*the boy wants the football*" e sua respectiva representação lógica:

$\exists$ *X, Y, C*
instance (X, want-01) $\wedge$
instance (Y, boy) $\wedge$
instance (Z, football) $\wedge$
ARG0 (X, Y) $\wedge$
ARG1 (X, Z)

A Smatch calcula a quantidade de conceitos entre duas sentenças utilizando precisão, revocação e medida-F de acordo com a quantidade de casamentos (*match*) obtidos. Uma dificuldade é que os nomes das variáveis podem ser diferentes entre as duas representações AMR. Para solucionar esse problema, conforme a Tabela 1 ilustra, a forma mais simples é realizar a troca do nome das variáveis entre todas as possíveis alterações na forma um-para-um e utilizar o maior valor para a medida-F como resultado da avaliação.

A Tabela 1[4] apresenta a avaliação entre as duas representações AMR formatadas em lógica proposicional apresentadas anteriormente, em que M é o total de casamentos entre as triplas das duas representações. Por exemplo, para a primeira linha, ocorram 4 casamentos, trocando o nome das variáveis, sendo que a única tripla para a qual não ocorreu o casamento foi a *instance(Z, football)* com *instance(C, go-01)*.

Existem formas mais elegantes para calcular e otimizar a execução da Smatch, como utilizar programação linear inteira e o método de *hill-climbing*, mas a ideia continua a mesma de avaliar a quantidade de conceitos em comum entre as duas representações.

[4] P é a precisão, R é a revocação e F é a medida-F.

Tabela 1 – Exemplo da avaliação de representações semânticas através da Smatch

			M	P	R	F
x=a	y=b	z=c	4	4/5	4/6	0.73
x=a	y=c	z=b	1	1/5	1/6	0.18
x=b	y=a	z=c	0	0/5	0/6	0.00
x=b	y=c	z=a	0	0/5	0/6	0.00
x=c	y=a	z=b	0	0/5	0/6	0.00
x=c	y=b	z=a	2	2/5	2/6	0.36
Smatch score:						0.73

Fonte: Retirado de Cai e Knight (2013)

Capítulo 4
TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados os principais trabalhos relacionados, direta ou indiretamente, com a GDRI de imagens naturais (não artificiais). Existem trabalhos que tratam especificamente da GDRI, produzindo uma representação em forma de texto a partir de uma região específica da imagem (ZHANG et al., 2015; KARPATHY; FEI-FEI, 2015), outros que utilizam o termo de geração de expressões de referência (MAO et al., 2016; YU et al., 2016), outros ainda que geram legendas densas para imagens (JOHNSON et al., 2016; YANG et al., 2017). Um outro trabalho importante de recuperação de informação também é apresentado em Karpathy et al. (2014). Apesar dos trabalhos se referirem a termos diferentes e algumas vezes utilizarem técnicas e abordagens diferentes, as estratégias apresentadas por eles podem ser adaptadas e utilizadas para GDRI.

4.1 Karpathy et al. (2014)

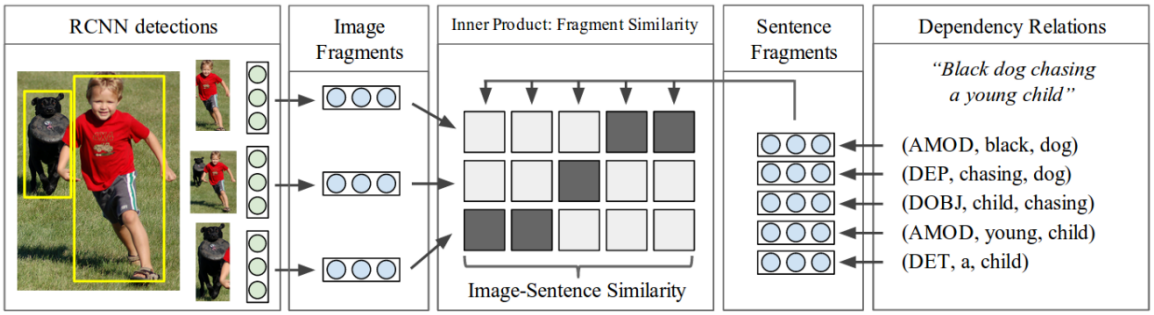
O trabalho de Karpathy et al. (2014) situa-se na área de recuperação de informação e tem como objetivo recuperar as imagens relevantes a partir de uma sentença de busca. O contrário também é válido, isto é, a partir da imagem busca-se recuperar sentenças relevantes.

O treinamento do modelo ocorre com um conjunto de imagens e sentenças correspondentes que descrevem os respectivos contextos. A abordagem utiliza a imagem e a sentença convertendo-os no que os autores chamam de fragmentos. A sentença é empregada como fragmentos utilizando uma árvore de dependência (gerada usando o Stanford CoreNLP[1]), em contrapartida à abordagem convencional de utilizar a sentença pura em língua natural. As relações de árvores de dependência são utilizadas para descrever as entidades da imagem. Como observado no exemplo da Figura 17, identificou-se duas entidades, (*dog* e *child*), seus respectivos atributos (*black* e *young*) e uma ação que os conecta (*chasing*).

Inspirado em trabalhos anteriores, os autores observaram que essas relações de dependência provêem uma informação mais rica (principalmente a relação entre as entidades) e eficiente

[1] https://stanfordnlp.github.io/CoreNLP/

Figura 17 – Similaridade de fragmentos entre imagem-sentença



Fonte: Retirado de Karpathy et al. (2014)

(do ponto de vista computacional) do que as próprias palavras ou bigramas.

A detecção de objetos como fragmentos se dá utilizando a RCNN, pré-treinada na ImageNet (DENG et al., 2009) e ajustada com 200 classes da *ImageNet Detection Challenge*, utilizando as 19 primeiras localizações e a imagem completa para computar o vetor de *embedding*. A RCNN (lado esquerdo da Figura 18) detecta os objetos e mapeia os fragmentos (no caso 3 fragmentos: o menino; o menino e o cachorro; e o cachorro) para um vetor de *embedding*. A árvore de dependência (lado direito) da sentença também é embutida em uma *embedding*. O modelo executa o produto interno entre os fragmentos (imagem e sentença) para obter a similaridade.

A Figura 18 exibe como ocorre o alinhamento de fragmentos entre os vetores da imagem e a sentença. As linhas representam fragmentos da imagem (v_i) e as colunas, da sentença (s_j). Cada quadrado é um resultado de $y_{ij} = sign(v_i^T s_j)$. As caixas vermelhas ocorrem quando $y_{ij} = -1$. As amarelas quando o fragmento da sentença ocorre em pelo menos um fragmento da imagem, enquanto o verde quando $y_{ij} = 1$. Ao final, os resultados são somados, o que pode ser entendido como um alinhamento.

Os experimentos foram conduzidos com 3 diferentes conjuntos de dados: Pascal1K (RASHTCHIAN et al., 2010), Flickr8K (HODOSH et al., 2013) e Flickr30K (YOUNG et al., 2014). Os conjuntos possuem 1.000, 8.000 e 30.000 imagens respectivamente anotadas com 5 sentenças geradas por anotadores diferentes por meio do *Amazon Mechanical Turk*[2] (AMT).

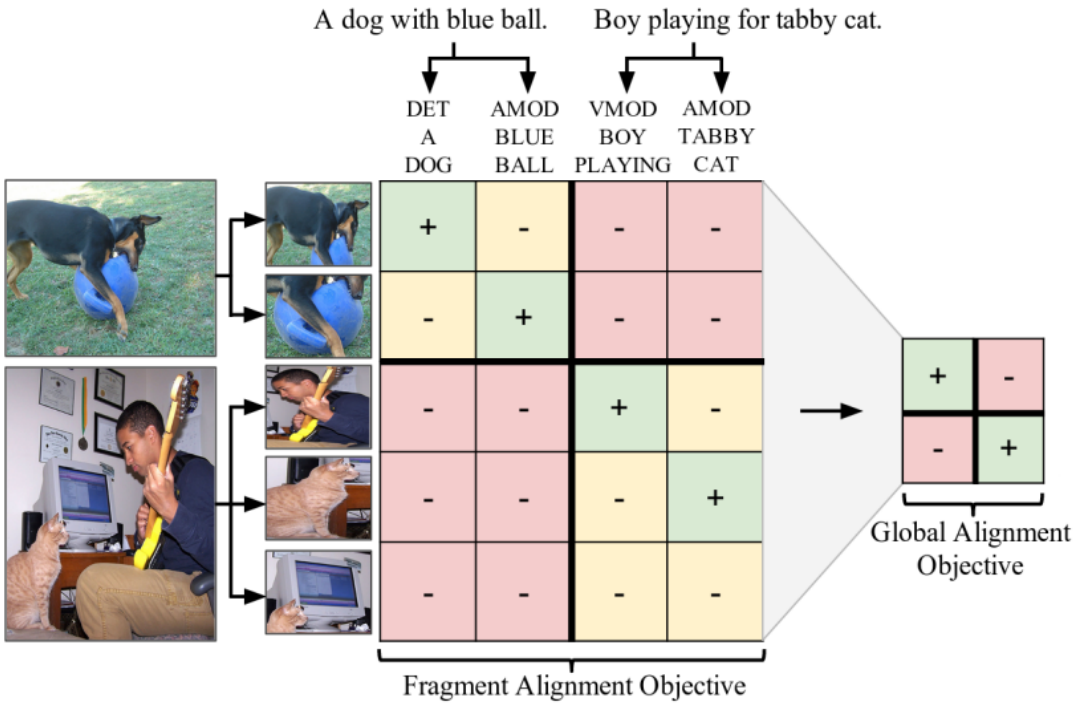
Para avaliar o modelo, foram considerados os valores de *mean rank*[3] e *recall*. O método proposto foi superior aos existentes para o Flick8k e Flickr 30k em ambas as tarefas (recuperação da imagem e recuperação da sentença). Já para o Pascal1K, o método proposto superou na recuperação da sentença e se mostrou competitivo na busca por imagem.

Apesar do bom desempenho, o método proposto apresenta limitações. Uma frase pode

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

[3] Medida utilizada junto com o teste de Kruskal–Wallis para testar se a classificação média é igual em todos os grupos.

Figura 18 – Alinhamento de vetores de fragmentos



Fonte: Retirado de Karpathy et al. (2014)

ser dividida em vários fragmentos, como em *black and white dog* que é convertida em dois fragmentos *(CONJ, black, white)* e *(AMOD, white, dog)*. Uma outra limitação é que muitas relações de dependência não são mapeadas, como *each other*, ou não modeladas, como a referência a várias entidades na sentença (*three children*). Outra limitação clara é que quando existe mais de uma ocorrência do objeto para a imagem, não é possível realizar a desambiguação por não suportar o uso de relações espaciais.

4.2 Karpathy e Fei-Fei (2015)

O trabalho de Karpathy e Fei-Fei (2015) propõe um modelo de alinhamento entre uma imagem e suas legendas com o objetivo de aprender a gerar descrições para regiões específicas da imagem.

O modelo é capaz de gerar tanto legenda para imagens como também descrições para regiões específicas. Quando utilizado para geração de legendas, primeiro é realizado o alinhamento de trechos da sentença que descreve a imagem por completo (semelhante a uma legenda) com as regiões. Essa sentença possui referências a vários objetos e regiões da imagem que serão quebradas em trechos que devem ser alinhados.

Para extrair trechos da legenda é utilizada uma Rede Neural Recorrente Bidirecional (*Bidirectional Recurrent Neural Network*, ou BRNN). Para isso, uma sequência de n palavras

é transformada em um vetor de h-dimensões, adicionando o contexto de tamanho variável em torno da palavra, usando um índice $t = 1...n$ para indicar a posição da palavra na sentença. A BRNN é definida formalmente como:

$$x_t = W_w \Pi_t \tag{4.1}$$

$$e_t = f(W_e x_t + b_e) \tag{4.2}$$

$$h_t^f = f(e_t + W_f h_{t-1}^f + b_f) \tag{4.3}$$

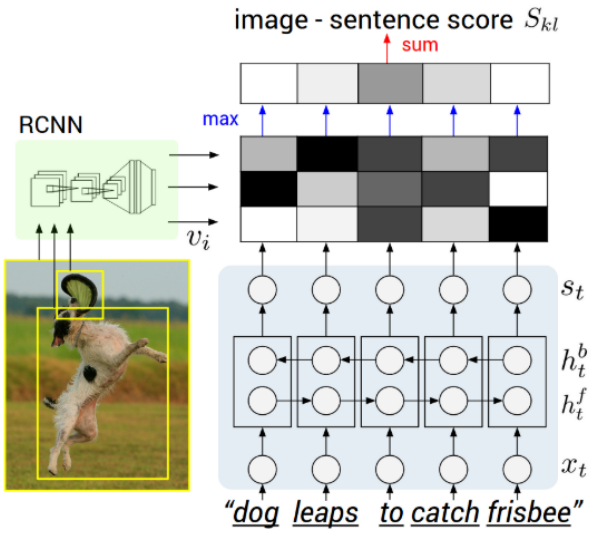
$$h_t^b = f(e_t + W_b h_{t+1}^b + b_b) \tag{4.4}$$

$$s_t = f(W_d(h_t^f + h_t^b) + b_d) \tag{4.5}$$

Os pesos W_w especificam uma matriz de *word embedding* inicializada com o word2vec (MIKOLOV et al., 2013) e mantida fixa para evitar *overfitting* e Π_t indica um vetor que tem um único índice da t-ésima palavra em um vocabulário de palavra.

Conforme ilustra a Figura 19, a BRNN consiste de dois fluxos independentes, um movendo para a direita (h_t^f) e outro para a esquerda (h_t^b). A representação final de h-dimensões representa s_t para a t-ésima palavra.

Figura 19 – Representação de uma sentença utilizando BRNN

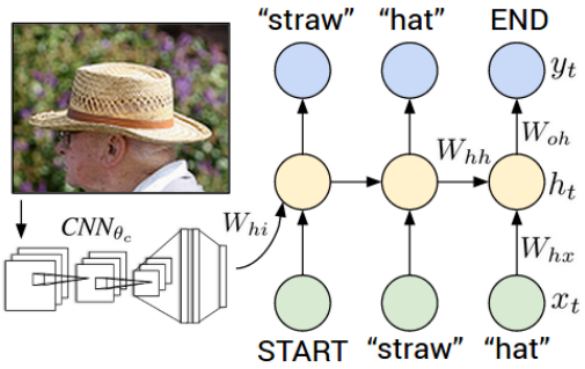


Fonte: Retirado de Karpathy e Fei-Fei (2015)

O modelo aprende todos os parâmetros W_e, W_f, W_b, W_d e seus respectivos *biases* b_e, b_f, b_d, b_d. A camada oculta varia entre 300 e 600 dimensões e como função de ativação usou-se a ReLU.

A geração, conforme apresenta a Figura 20, ocorre através de uma Rede Neural Recorrente Multimodal (*Multimodal Recurrent Neural Network*, ou MRNN), que recebe esse nome pois tem como uma de suas entradas a última camada da rede CNN. O tamanho da camada oculta é de 512 neurônios. A BRNN é treinada para combinar a palavra com o contexto e predizer a próxima palavra.

Figura 20 – Representação da MRNN



Fonte: Retirado de Karpathy e Fei-Fei (2015)

Para gerar descrições para regiões de imagens, foi anotado um conjunto com o auxílio do *Amazon Mechanical Turk* (AMT) a partir do *Microsoft Common Objects in Context*[4] (MSCOCO) (KRISHNA et al., 2017). Uma imagem foi exibida para 9 pessoas diferentes para delimitar e informar o rótulo de 5 *bbox*. Foram coletadas 9.000 descrições para 200 imagens (média de 45 por imagem) sendo que cada descrição continha em média 2,3 palavras.

A avaliação ocorreu utilizando a medida BLEU, sendo que o modelo foi superior quando treinado com as regiões do que quando treinado com a legendas, avaliando-o com regiões. Os valores obtidos foram: BLEU-1 0,352; BLEU-2 0,23; BLEU-3 0,16; e BLEU-4 0,148. Como *baseline*, utilizou-se o *Nearest Neighbor*, obtendo 0,229 para BLEU-1, 0,105 para BLEU-2 e 0 para BLEU-3 e BLEU-4.

Como o conjunto criado para a descrições das regiões foi consideravelmente menor (200 imagens) do que para legenda de imagens, não é possível concluir que o modelo proposto foi realmente melhor, devido a quantidade de poucos exemplos com os demais trabalhos aqui apresentados.

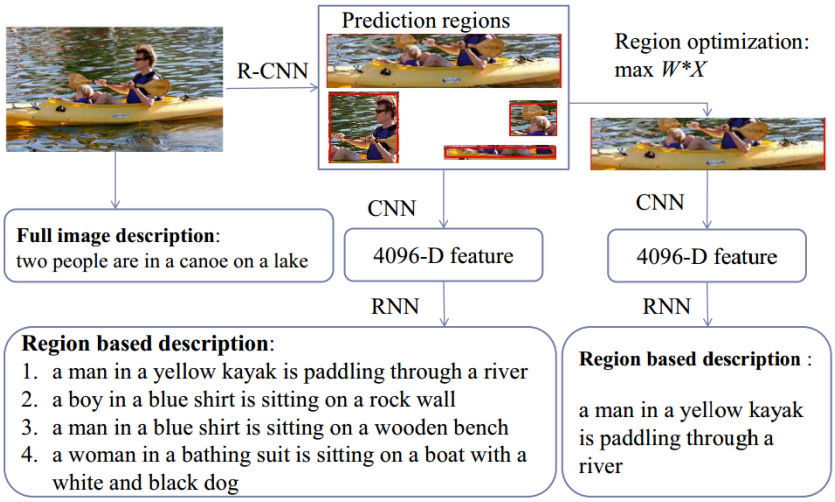
[4] http://cocodataset.org/

4.3 Zhang et al. (2015)

O trabalho teve como objetivo produzir descrições para regiões da imagem. O modelo proposto é composto pela RCNN (GIRSHICK et al., 2014), uma RNN do tipo LSTM e um método de otimização para reduzir o número de regiões selecionadas.

Para a RCNN foi empregada a arquitetura VGG-16 (SIMONYAN; ZISSERMAN, 2015) para extrair as características. Para reduzir o número de regiões propostas, utilizou-se uma otimização, utilizando $X = \{x_1, x_2, x_3, x_4\}$ como os critérios de seleção da região, nos quais x_1 é o tamanho da área, x_2 é o centro da região, x_3 é a universalidade do rótulo e x_4 é a pontuação da predição da sentença e $W = w_1, w_2, w_3, w_4$ o parâmetro otimizador. A função objetivo foi definida como $Y = maxW * X$, sendo Y a pontuação final das descrições das regiões. A arquitetura completa é ilustra na Figura 21.

Figura 21 – Exemplos de predição de Zhang et al. (2015)



Fonte: Retirado de Zhang et al. (2015)

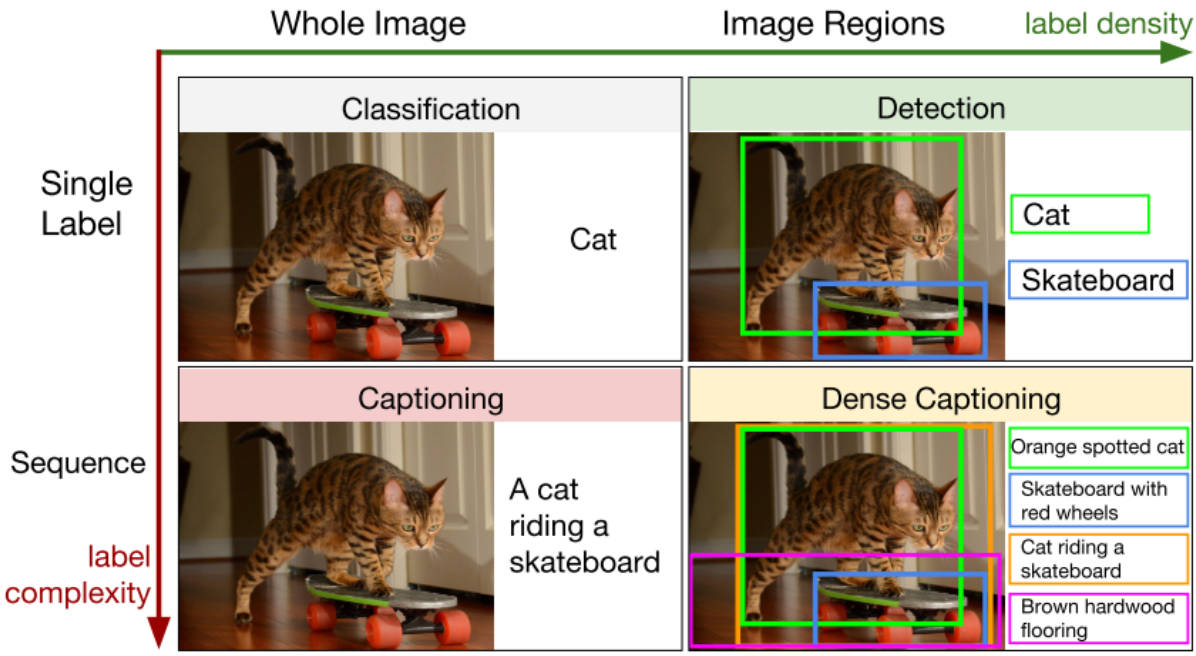
Para o treinamento, foi usado o conjunto de dados ImageNet (DENG et al., 2009) empregado na RCNN, enquanto os conjuntos Flickr8K e o MSCOCO (KRISHNA et al., 2017) foram utilizados para treinamento da LSTM. A partir das características da região da imagem (R), proveniente da RCNN é o vetor de entrada ($x_1, ...x_t$) para a LSTM, que computa a sequência de estados ($h_1, ..., h_t$), obtendo a sequência de resultados ($y_1, ..., y_2$), iterando a relação de $t = 1...N>$

$$x_{-1} = RCNN(R) \tag{4.6}$$

$$h_t = LSTM(h_{t-1}, x_{t-1}), t \in \{1, ..., N\} \tag{4.7}$$

$$y_t = softmax(h_t, x_t), t \in \{1, ..., N\} \tag{4.8}$$

Figura 22 – Definição de legenda densa



Fonte: Retirado de Johnson et al. (2016)

Os experimentos foram conduzidos em ambos os *datasets*, Flickr8K e MSCOCO. Para o Flickr, utilizou-se o particionamento disponível, enquanto para o MSCOCO 1.000 exemplos foram utilizados para teste.

Como métrica, optou-se por utilizar a BLEU, com n-grama variando de 1 a 4. Para o Flickr, os resultados foram: BLEU-1 0,466835, BLEU-2 0,25327, BLEU-3 0,102291, BLEU-4 0,040732; enquanto para o MSCOCO foram: BLEU-1 0,506205, BLEU-2 0,274891, BLEU-3 0,107493 e BLEU-4 0,046179.

4.4 Johnson et al. (2016)

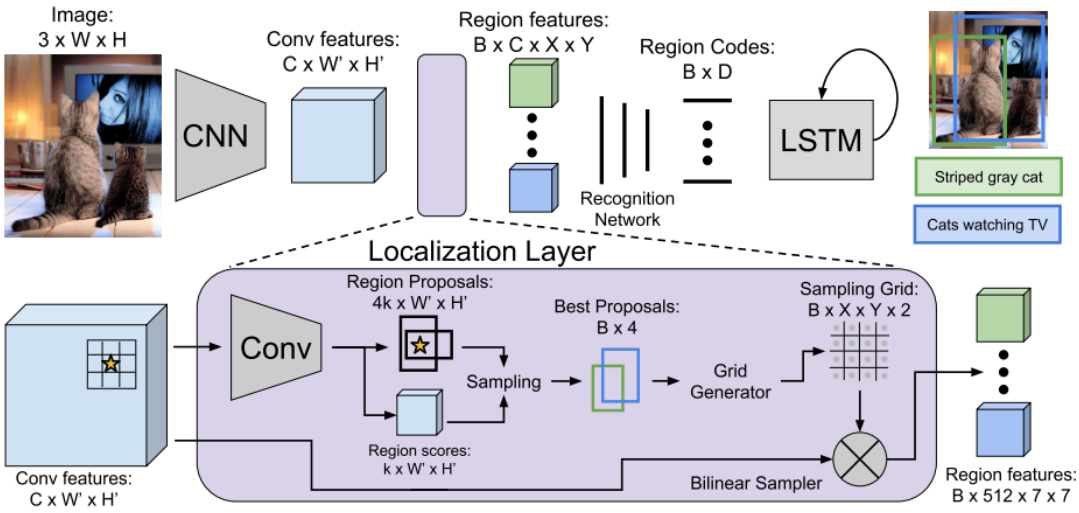
Johnson et al. (2016) desenvolveram um modelo de geração de legendas densas. O problema de gerar legendas densas (*dense captioning*) pode ser entendido como uma tarefa semelhante, se não a mesma, que a GDRI. É a generalização da detecção de objetos, quando a descrição do objeto consiste em uma palavra ou conjunto pré-definido de palavras (JOHNSON et al., 2016). O termo possui o substantivo "legenda" por ser capaz de descrever um conjunto de regiões e objetos, se assim necessário, algumas vezes até intersectando ou contendo uma outra legenda. O adjetivo "denso" deve-se à maior riqueza de informação. A Figura 22 ilustra o que seria uma legenda densa. Nessa figura, o eixo horizontal cresce conforme aumenta a densidade do rótulo, isto é, de acordo com a granularidade (quantidade de regiões) da imagem. O mesmo acontece com o eixo vertical, em que o rótulo aumenta a sua densidade de acordo com o nível de detalhamento.

Johnson et al. (2016) usaram uma Rede de Localização Totalmente Convolucional (*Fully Convolutional Localization Network*, ou FCLN) para a tarefa de geração e compreensão legendas densas. A FCLN é composta por uma rede CNN e RNN, assim como os outros trabalhos da atualidade. Contudo, foi inserida uma camada de localização densa na CNN, que prediz um conjunto de regiões de interesse.

A Figura 23 ilustra o processo de treinamento desta rede no qual a rede CNN utilizada é a VGG-16 (SIMONYAN; ZISSERMAN, 2015), que consiste de 13 camadas de 3x3 intercaladas com 5 camadas de 2x2 de *max pooling*. A última camada de *max pooling* é retirada, então uma imagem de tamanho 3 x W x H é origem para um tensor[5] de características de C x W' x H', onde C = 512, $W' = [\frac{W}{16}]$ e $H' = [\frac{H}{16}]$. A saída é a imagem codificada que é a entrada da camada de localização. A camada de localização recebe um tensor de entrada, identifica regiões espaciais de interesse e extrai uma representação de tamanho fixo de cada região. A camada de localização seleciona B regiões de interesse e tem como saída três tensores de informação sobre as regiões, compostos por:

- Coordenadas das regiões: Uma matriz de B x 4 com as coordenadas de cada uma das *bounding boxes*.
- Pontuação das regiões: Um vetor de tamanho B com a pontuação de cada uma das regiões. Regiões com alta pontuação correspondem a regiões de maior interesse.
- Características das regiões: Um tensor B x C x X x Y que possui características das regiões em uma grade X x Y de C-dimensões.

Figura 23 – Treinamento FCLN



Fonte: Retirado de Johnson et al. (2016)

[5] Objetos matemáticos que descrevem propriedades físicas como escalares e vetores.

Essa abordagem é semelhante a Faster-RCNN (Ren et al., 2017), exceto pelo fato de o RoI ser substituído por uma interpolação bilinear, a retro-propagação do gradiente e também das regiões de interesse detectadas. A interpolação bilinear é a extensão na interpolação linear para interpolar (estimar um valor aproximado para uma função) em uma grade regular de duas variáveis. Em geral, são usados polinômios para descobrir os pontos discretos.

Para geração da sentença em língua natural, utilizou-se uma RNN do tipo LSTM. A partir de sequência de *tokens* $s_1, ...s_t$, a entrada da LSTM é $T+2$ de vetores de palavras $x_{-1}, x_0x_1, ...x_T$, no qual $x_{-1} = CNN(I)$ é a região codificada e x_0 é o *token* especial de início (*START*).

Os experimentos foram realizados com o conjunto de dados Visual Genome[6], contendo 94.313 imagens e 4.100.413 descrições para as regiões (média de 43,5 regiões por imagem) com a intersecção de exemplos com o conjunto *Yahoo Flickr Creative Commons 100 Million* (YFCC100M) (THOMEE et al., 2016). Contudo, os autores não detalham qual foi o tamanho resultante desta intersecção.

O pré-processamento ocorreu trocando as palavras que ocorriam menos de 15 vezes pelo *tokens* especial <*UNK*>, resultando em um vocabulário de 10.497 palavras. Frases como *there is a* e *this seems to be a*, descrições com mais de 10 palavras, e imagens com menos de 20 e mais de 50 descrições, foram descartadas. O pré-processamento resultou em 87.398 imagens, sendo que 5.000 foram destinadas para validação, 5.000 para teste, e o restante para treinamento.

Durante a validação e teste, as *bouding boxes* também foram pré-processadas para que sobreposições fossem consideradas uma única região. A *bbox* com maior número de sobreposições foi combinada (utilizando a média) em uma única com várias descrições.

Para avaliação da qualidade da sentença gerada pelo modelo, utilizou-se o mAP com o METEOR e variações de limiares entre 0, 0,5, 0,1, 0,15, 0,2 e 0,25. A avaliação das descrições também ocorre com as descrições das *bbox* que foram mescladas, conforme descrito anteriormente, obtendo 0,273 para o METEOR e 5,39 para o mAP.

4.5 Mao et al. (2016)

O trabalho de Mao et al. (2016) propõe um novo método para o que definem como geração e compreensão de expressões de referência, utilizando técnicas de RNA e *deep learning* para expressões de referência ambíguas. Segundo Viethen e Dale (2006), geração de expressões de referência (GER) é o processo de identificação e seleção de propriedades úteis para descrever o objeto das outras entidades do contexto. Dessa forma, como empregado por Mao et al. (2016), pode ser entendido como uma tarefa semelhante a GDRI.

Formalmente, Mao et al. (2016) definem a GER como $argmax_s p(S|R, I)$, no qual S é a sentença, R a região e I a imagem completa. Uma RNN do tipo LSTM é usada para representar

[6] https://visualgenome.org/

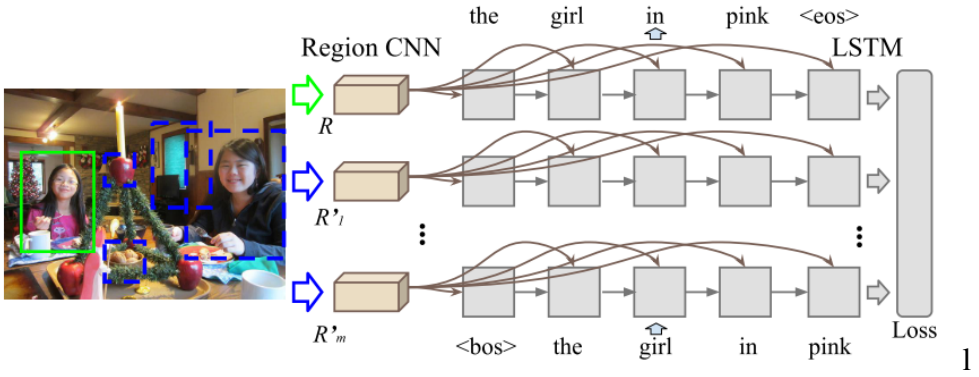
$argmax_s p(S|R, I)$, onde em S é gerada uma palavra por vez até gerar um fim de sentença. Para computar a sentença mais provável, foi utilizada a busca por feixe de tamanho 3. Esse método é semelhante a outros de legenda de imagem, exceto pelo fato que utiliza a região e não a imagem completa. O método $p(S|R, I)$ é comum para tarefa de legenda de imagem com uma CNN e LSTM, mas não tem como finalidade distinguir sentenças.

O método de treinamento é dado por :

$$MI(S, R) = \log \frac{p(S, R)}{p(R)p(S)} = \log \frac{p(S|R)}{P(S)} \tag{4.9}$$

no qual $p(S) = \Sigma_{R'}\ p(S|R')p(R') = \Sigma_{R'}\ p(S|R')$. Esse método é chamado de Informação Mútua Máxima (*Maximum Mutual Information*, ou MMI) (MAO et al., 2016). O método penaliza uma expressão de referência para um objeto se ela também for provável para outro objeto da mesma imagem. A Figura 24 ilustra o processo de execução de maneira simplificada, onde R (cor verde) é a região alvo e R' é a região incorreta. O modelo procura um valor de $p(S|R, I)$ maior que $p(S|R', I)$ se $R' \neq R$.

Figura 24 – Execução para regiões corretas e incorretas utilizando MMI



Fonte: Retirado de Mao et al. (2016)

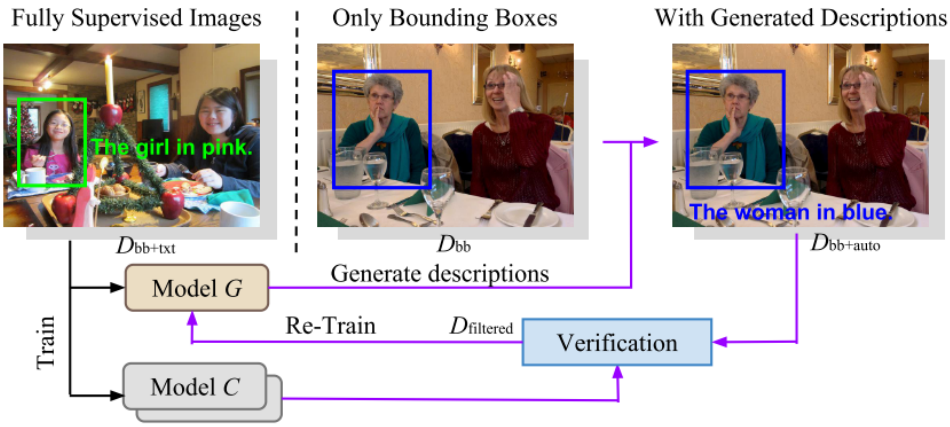
Foram propostos dois métodos de função objetivo para combinar com o método de treinamento MMI. Um foi uma variação da *softmax* e a outra a Margem Máxima (*Max-Margin*, ou MM), sendo que esse último obteve melhor resultado, sendo definido por :

$$J''(\theta) = -\sum_{n=1}^{N}\{\log\ p(S_n|R_n, I_n, \theta) - \lambda\ max(0, M - \log\ p(S_n|R_n, I_n, \theta) + \log p(S_n|R'_n, I_n, \theta))\} \tag{4.10}$$

O símbolo θ refere-se aos parâmetros da RNN e CNN, somados os exemplos de treinamento. Utilizou-se gradiente estocástico com tamanho do *batch* de 16 e taxa de aprendizado inicial de 0,01 reduzida a metade a cada 50.000 iterações. Para evitar *overfitting*, é utilizado um *dropout* de 0,5 na LSTM.

O treinamento ocorreu de forma semi supervisionada, conforme ilustra a Figura 25. Uma parte dos dados rotulados ($D_{bbox+txt}$) possuía as *bbox* e as descrições (txt) junto com uma outra parte que possuía apenas as *bbox* (D_{bbox}) sem as descrições. O modelo G foi treinado com os dados $D_{bbox+txt}$ para computar $p(S|R, I)$ para gerar as descrições para D_{bbox} criando os dados $D_{bbox+auto}$. O modelo então foi retreinado com $D_{bbox+txt} \cup D_{bbox+auto}$ com aprendizado *bootstrap*. No entanto, nem todas as sentenças geradas por $D_{bbox+auto}$ são corretas. Por isso ocorre um filtro para decodificar cada sentença de $D_{bbox+auto}$ e são mantidas as que mapeiam para o objeto correto ($D_{filtered}$) . Este processo minimiza a chance de ocorrer *overfitting* e ainda aumenta a variedade de diferentes modelos que podem ser gerados.

Figura 25 – Representação da construção do conjunto de dados semi supervisionado



Fonte: Retirado de Mao et al. (2016)

Os experimentos foram conduzidos com os conjuntos G-Ref e com o UNC-Ref. O G-Ref possui 49.820 objetos para 27.799 imagens, uma média de 1,93 objetos por imagem. O total de expressões de referência disponíveis é de 95.010, sendo em média 1,90 expressões de referência para cada objeto e 3,69 expressões de referência para cada imagem. O UNC-Ref possui 50.000 objetos identificados para 19.994 imagens. O particionamento ocorreu selecionando 5.000 objetos para validação e 5.000 para teste em cada um dos conjuntos. O restante foi designado para treinamento.

A avaliação ocorreu de duas formas. Na primeira, utilizou-se o AMT para perguntar às pessoas, dadas duas sentenças (uma gerada pelo modelo e outra gerada por um humano), qual sentença era melhor ou se as duas eram suficientemente boas para identificar uma *bbox*. No entanto, como esse tipo de avaliação não é escalável, foi realizada também uma avaliação *end-to-end*, na qual a partir de uma região da imagem gera-se a sentença que é a entrada para o modelo de compreensão, o qual deve ser capaz de identificar a região original.

O modelo obteve melhor resultado quando treinado de modo supervisionado, obtendo o valor de 0,833 avaliando-o na forma *end-to-end*. Para a avaliação humana 1.000 objetos e suas ERS foram apresentadas para pessoas no AMT. Sendo que 20,4% responderam que a ERS

geradas pelo modelo eram tão boas ou melhores do que a referência (previamente anotada).

Como contribuição, além do modelo que superou o estado da arte, também está disponível o conjunto de dados G-Ref, podendo ser utilizado por outros pesquisadores para comparação dos resultados.

4.6 Yu et al. (2016)

O trabalho de Yu et al. (2016) aborda a tarefa de geração e compreensão de expressões de referência, que tem por objetivo produzir e melhorar expressões de referência não ambíguas, assim como Mao et al. (2016). Para isso, utiliza uma técnica de comparação visual de objetos de mesma categoria pertencentes a imagem. Os autores justificam que essa abordagem é semelhante a humana que quando descreve um objeto através de uma expressão de referência utiliza as mesmas características para os outros objetos da categoria da imagem. Para isso utilizou-se duas técnicas: comparação visual e geração em língua natural de maneira conjunta.

Para comparação visual, primeiro é calculada a diferença entre o objeto alvo e os objetos da mesma categoria presentes na imagem através das características da CNN. A diferença é calculada subtraindo cada um dos objetos com o objeto alvo. Para representar a diferença visual entre o objeto alvo e os outros objetos, é calculado um vetor agregado como representação com a média sobre cada dimensão de característica. Depois é realizada a codificação da relação relativa entre até 5 objetos da mesma categoria e o objeto alvo, com o intuito de utilizar as propriedades relativas e as relações espaciais.

A geração em língua natural de maneira conjunta é realizada gerando uma expressão de referência para todos os objetos da mesma categoria, ao contrário das outras abordagens que geram para cada um dos objetos de forma independente. Essa geração assemelha-se ao processo humano de utilizar as mesmas características com valores diferentes para uma mesma categoria de objeto. Por exemplo, em uma imagem que possui dois carros, se para um carro é utilizada uma frase como “o carro preto”, para o outro carro o ser humano tende a utilizar a cor para diferenciação, como “o carro branco”. No entanto, se um valor do atributo foi utilizado em uma expressão de referência, ele não pode ser utilizado para outro objeto da mesma categoria presente na imagem.

Conforme ilustrado na Figura 26, o modelo utiliza uma LSTM que possui, além da memória convencional, também uma “conexão” entre expressões para objetos diferentes, a qual gera múltiplas expressões de referência, $\{r_i\}$, dados os objetos da mesma categoria, $\{o_j\}$.

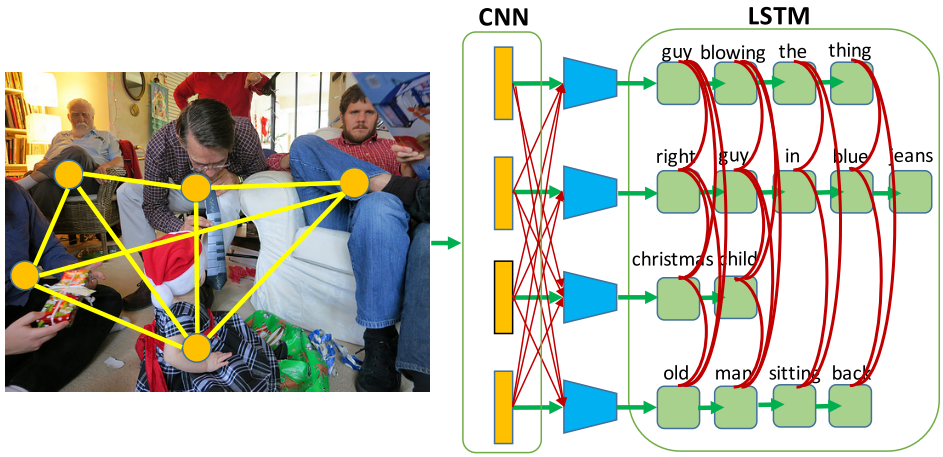
$$P(R|O) = \prod_i P(r_i|o_i, \{o_j \neq i\}, \{r_j \neq i\}), \; = \prod_i \prod_t P(w_{i_t}|w_{i_{t-1}}, ..., w_{i_1}, v_i, \{h_{j_t, j \neq i}\}) \tag{4.11}$$

onde w_{i_t} são as palavras no tempo t, v_i as representações visuais, e h_{j_t} é a saída do j-ésimo objeto no tempo t que codifica informações visuais e da sentença para o j-ésimo objeto. Como comparação visual, a diferença da saída é agregada para dificultar informações ambíguas.

$$P(w_{i_t}|w_{i_{t-1}}, ..., w_{i_1}, v_i, \{h_{j_t, j \neq i}\}) = softmax(W_h[h_{dif_{i_t}}] + b_h) \tag{4.12}$$

em que $h_{dif_{i_t}} = \frac{1}{n} \sum_{j \neq i} \frac{h_{i_t} - h_{j_t}}{||hi_t - h_{j_t}||}$ e n é o número de objetos da mesma categoria. A diferença é incorporada com a saída do objeto e encaminhada à *softmax* para prever a palavra.

Figura 26 – Geração de ERs conjunta com conexões entre si



Fonte: Retirado de Yu et al. (2016)

Os experimentos foram conduzidos com 3 conjuntos de dados: o G-Ref de Mao et al. (2016), o RefCOCO e o RefCOCO+. Esses dois últimos foram criados a partir do jogo ReferItGame (KAZEMZADEH et al., 2014). Nesse jogo, entre dois participantes, o primeiro jogador recebia a imagem e um objeto para descrever em língua natural para que o segundo jogador, através da imagem e da descrição, identifica-se o objeto. A dupla que acertasse mais objetos ganhava. As imagens foram selecionadas para que tivessem 2 ou mais objetos da mesma categoria. Para o RefCOCO, nenhuma restrição de linguagem foi utilizada, enquanto para o RefCOCO+ os jogadores foram recomendados a não utilizar palavras de localização produzindo, assim, um conjunto com descrições focadas em propriedades do objeto alvo e não em propriedades relativas e relações espaciais.

O RefCOCO possui 142.209 expressões (média de 3,61 palavras) para 50.000 objetos em 19.994 imagens, enquanto RefCOCO+ possui 141.564 expressões (média de 3,53 palavras) para 49.856 objetos em 19.992 imagens. RefCOCO+ e RefCOCO possuem uma média de 3,9 objetos da mesma categoria para cada imagem.

Esses conjuntos de dados foram divididos de duas maneiras para a realização dos experimentos. Na primeira, a divisão entre treino e teste foi realizada considerando-se os objetos, assim como Mao et al. (2016), para o G-Ref. Mais especificamente, no G-Ref, o objeto de uma

determinada categoria que ocorre na imagem vai para o conjunto de treinamento e outros objetos da mesma categoria que ocorrem na imagem podem ir para o conjunto de teste ou de validação. Já o RefCOCO e RefCOCO+ foram divididos da forma convencional (por imagem). A segunda divisão foi realizada com base nas categorias dos objetos, para o RefCOCO e RefCOCO+, em: pessoas (teste A) e outros objetos (teste B). Essa divisão ocorreu após os autores observarem que quase metade das referências nesses dois conjuntos de dados eram para pessoas. Para essa divisão, todos os objetos da mesma imagem foram colocados em uma única partição (treino ou teste).

O trabalho Mao et al. (2016) (descrito na Seção 4.5) é utilizado como *baseline*, sendo que foi avaliada a inserção de informações de contexto. A avaliação das expressões de referência geradas ocorreu utilizando a BLEU (unigrama e bigrama), ROUGE e METEOR, com os testes A e B para o RefCOCO e RefCOCO+, e o G-Ref como validação.

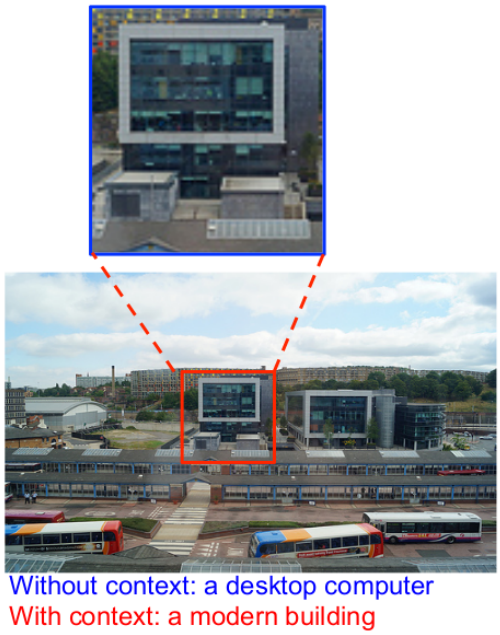
O modelo proposto com comparação visual e geração conjunta foi superior em todas as medidas e partições para o RefCOCO e RefCOCO+, exceto para o Teste A com BLEU-2, em que apenas a comparação visual foi levemente superior (0,318 contra 0,322), inclusive incorporando a geração conjunta para o modelo de Mao et al. (2016). Para o G-Ref não foi possível utilizar a geração conjunta visto que objetos de uma mesma imagem podem estar presentes em partições diferentes. Desta forma o modelo com apenas a comparação visual se mostrou superior também para o G-Ref em comparação ao de Mao et al. (2016) (0,227 contra 0,273 no melhor resultado utilizando a BLEU-2). Para a METEOR, obteve-se um resultado de 0,151 contra 0,149 de Mao et al. (2016).

Na avaliação com juízes humanos, na qual três pessoas no AMT tinham que clicar no objeto correto a partir da imagem e da ER gerada pelo modelo e considerava-se como correto quando no mínimo duas pessoas clicassem no respectivo objeto, o modelo mostrou-se apenas competitivo.

4.7 Yang et al. (2017)

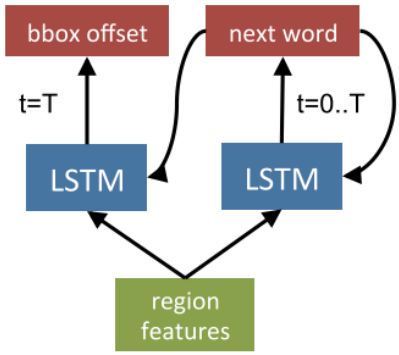
O trabalho de Yang et al. (2017) situa-se na tarefa de legendas densas, assim como o de Karpathy et al. (2014) e Johnson et al. (2016). O objetivo é produzir legendas densas com inferência conjunta e informações de contexto. Para isso, os autores combinam uma rede CNN e uma LSTM, uma abordagem similar aos trabalhos relacionados. No entanto, o trabalho de Yang et al. (2017) combina mais de uma LSTM para gerar a legenda. Assim, o modelo proposto é constituído de duas partes. A primeira parte é a inferência conjunta, em que a *bbox* predita pelo modelo é ajustada utilizando uma RNN a partir da informação da legenda gerada. A segunda é a combinação de contexto, onde as regiões preditas pela CNN são combinadas com as características da imagem completa para que o contexto auxilie (ou algumas vezes até altere) a descrição da região de interesse (Figura 27).

Figura 27 – Alteração da legenda através do contexto



Fonte: Retirado de Yang et al. (2017)

Figura 28 – Inferência conjunta para localização precisa



Fonte: Retirado de Yang et al. (2017)

Figura 29 – Exemplo da aplicação da inferência conjunta

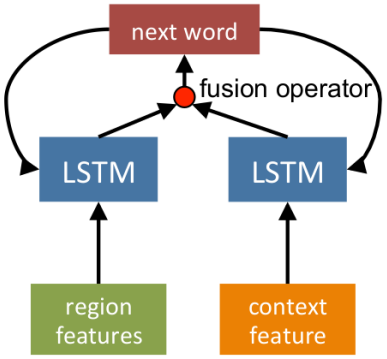
Fonte: Retirado de Yang et al. (2017)

A etapa de inferência conjunta é ilustrada na Figura 28. O ajuste da *bbox* é feito agora utilizando também uma LSTM, assim como a geração da sentença. As duas redes são chamadas de LSTM-localização e LSTM-legenda e ambas recebem como entrada a representação da última palavra. O ajuste da *bbox* ocorre após a geração da última palavra da sentença, quando a próxima palavra corresponder a um *token* de fim de sentença. Esse processo de ajuste é ilustrado na Figura 29.

No início da geração da legenda (*token* <*start*>) a *bbox* é marcada na cor laranja mais escuro. A cada passo, ocorre o ajuste (exemplificado com tons de laranja mais claros na Figura 29) até ocorrer o fim da sentença.

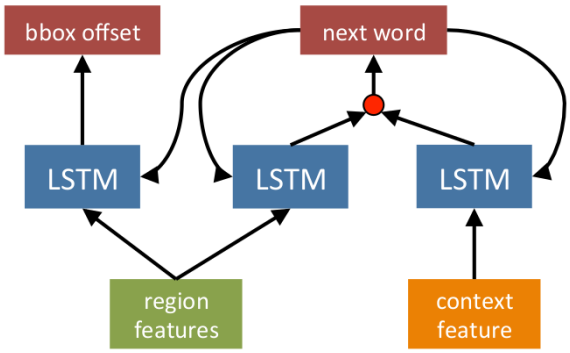
O deslocamento da *bbox* não está diretamente relacionado com o recurso de contexto, sendo que esse último é o único que irá auxiliar na previsão da legenda. No entanto, o contexto possui influência no ajuste da *bbox* através da inferência conjunta.

Figura 30 – Combinação do contexto com a região



Fonte: Retirado de Yang et al. (2017)

Figura 31 – Combinação do modelo de contexto com inferência conjunta



Fonte: Retirado de Yang et al. (2017)

O contexto é modelado utilizando uma LSTM-contexto, que é combinada (*fusion operator*) com a saída da LSTM da legenda, produzindo a próxima palavra (Figura 30). A próxima palavra servirá como entrada tanto para a LSTM-região quanto para a LSTM-contexto.

A integração dos modelos de combinação de contexto e de inferência conjunta ocorre conforme ilustrado na Figura 31. O treinamento ocorre minimizando a função de perda L, dada por:

$$L = L_{cap} + \alpha L_{det} + \beta L_{bbox} \tag{4.13}$$

na qual L_{cap} é a função de perda da legenda, L_{bbox} da *bbox* e L_{det} da detecção de objetos, com $\alpha = 0.1$ e $\beta = 0.01$. L_{cap} é a entropia cruzada[7] para previsão de cada palavra em cada tempo, L_{det} é a entropia da classe de primeiro e segundo plano e L_{bbox} é uma função de perda (Ren et al., 2017).

Durante os experimentos, os autores escolheram utilizar o Visual Genome V1.0 e a métrica mAP, ambos assim como Johnson et al. (2016), utilizando os mesmos limiares do METEOR reportados. A taxa de aprendizado inicial foi de 0,001 e decaiu a metade a cada 100 mil iterações, com um momento de 0,98. No pré-processamento, foram retiradas descrições com mais de 10 palavras. O resultado foi consideravelmente melhor que o de Johnson et al. (2016)

[7] É utilizada para medir o grau de desordem (confusão) em um conjunto.

(5,39 contra 9,31) e um resultado ainda melhor quando avaliado com o Visual Genome 1.2 (9,96). Os autores discutem que o melhor resultado na versão 1.2 se deve ao fato das legendas serem mais limpas.

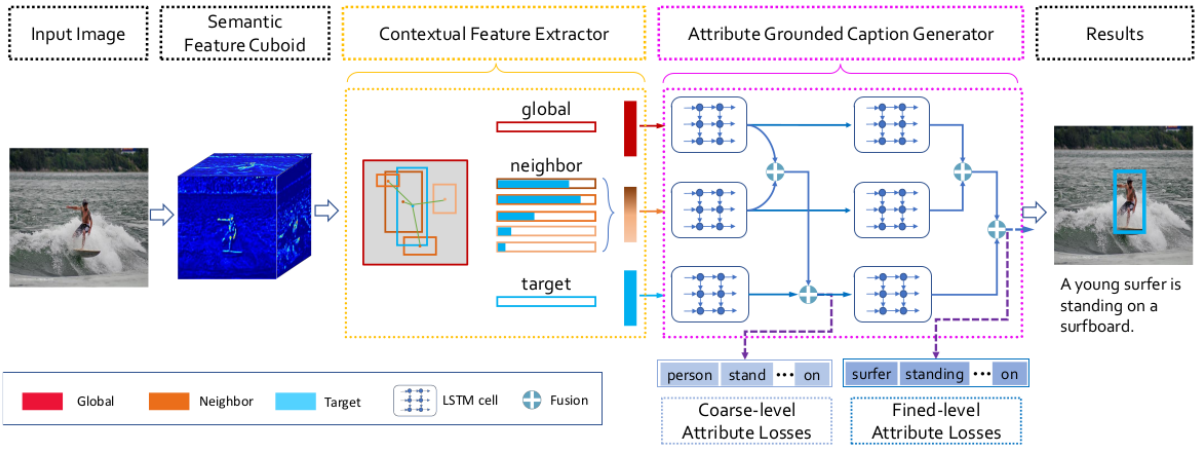
Após uma experimentação das operações com os conjuntos de dados, os autores chegaram à conclusão que a combinação (*fusion operator*) que corresponde ao melhor resultado é a multiplicação ($\otimes$) e concatenação $(.,.)$ para os conjuntos Visual Genome 1.0 e 1.2, respectivamente. No entanto, não é possível afirmar que os resultados são realmente melhores, em virtude de Johnson et al. (2016), durante o pré-processamento, selecionar apenas legendas entre 20 e 50 palavras, contra no máximo 10 de Yang et al. (2017).

4.8 Yin et al. (2019)

O trabalho de Yin et al. (2019) está situado na área de *dense captioning*, assim como os trabalhos de Johnson et al. (2016), Yang et al. (2017) e Zhang et al. (2019), sendo atualmente o estado da arte. A diferença para os demais trabalhos é que além de utilizar as informações de contexto para produzir uma *bbox* melhor, assim como Yang et al. (2017) e Zhang et al. (2019), são utilizadas informações dos vizinhos próximos da *bbox* alvo, propondo assim legendas densas baseadas em contexto e atributo (*Context and Attribute Grounded Dense Captioning* – CAG-Net).

A arquitetura completa é ilustrada na Figura 32. A imagem completa, dada como entrada, é convertida em uma estrutura que permite a fácil identificação dos vizinhos e relação espacial entre as regiões da imagem (*Semantic Feature Cuboid*). Logo depois, a arquitetura possui as duas principais partes: extração de características contextuais e gerador de legenda baseado em atributo.

Figura 32 – Arquitetura de Yin et al. (2019)



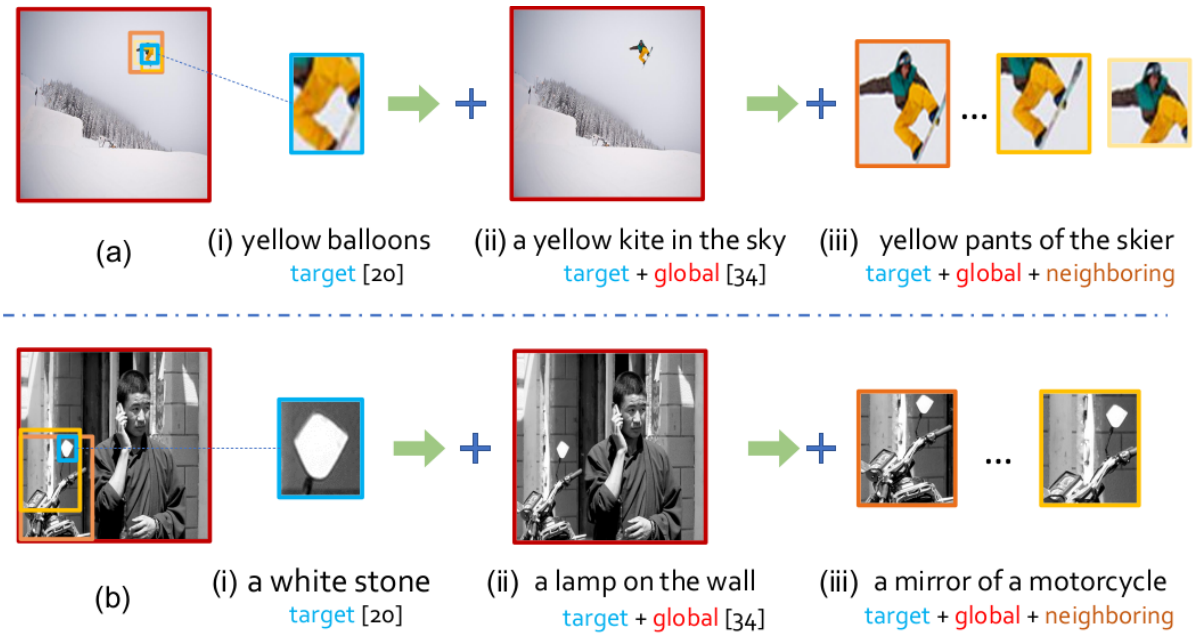
Fonte: Retirado de Yin et al. (2019)

A extração de características contextuais acontece a partir da estrutura construída anterior-

mente, em formato de cubo, para cálculo da similaridade e interação entre a RoI alvo e as RoI dos vizinhos, com base na similaridade e proximidade espacial. Isso permite o compartilhamento de informações contextuais adaptáveis de vários RoI adjacentes (globais e vizinhos) para interagir com o RoI alvo. A extração de características acontece com a *Faster RCNN* (Ren et al., 2017).

Para a geração de descrição baseada em atributo, é utilizada a arquitetura na forma de LSTM, em que são usadas 3 LSTM com propósitos diferentes: alvo, vizinhos e global. A LSTM alvo, como o próprio nome sugere, é responsável pela informação da RoI alvo. A LSTM global permite utilizar outras informações presentes na imagem, enquanto a LSTM vizinhos mapeia os objetos próximos ao alvo. Os autores justificam que a utilização de mais de uma LSTM pode aumentar o poder de representatividade. Um exemplo de uma predição da CAG-Net é ilustrado na Figura 33, em que informações vizinhas ao alvo auxiliam na produção de uma melhor descrição.

Figura 33 – Exemplo da Predição de Yin et al. (2019)



Fonte: Retirado de Yin et al. (2019)

Os experimentos foram realizados com o *Visual Genome*. O particionamento ocorreu destinando 5.000 exemplos para as partições de validação e teste cada, e 77.398 para treinamento. Foram mantidas apenas as 10.000 palavras mais frequentes e sentenças com mais de 10 palavras foram descartadas.

A avaliação do modelo ocorreu através das medidas METEOR e mAP, superando os modelos de Johnson et al. (2016) e Yang et al. (2017) em ambas, com um valor de 0,279 e 10,51, respectivamente.

4.9 Zhang et al. (2019)

Zhang et al. (2019) desenvolveram um trabalho na área de *dense captioning*, assim como os trabalhos de Johnson et al. (2016), Yang et al. (2017) e Yin et al. (2019), inclusive comparando seus resultados com os desses demais trabalhos. A diferença para os outros trabalhos é a utilização de um método de extração precisa de características (*Precise Feature Extraction* - PFE), tendo como motivação o fato de que Johnson et al. (2016) e Yang et al. (2017) buscam melhores resultados fazendo modificações em suas arquiteturas (sobretudo das RNA) e não dando importância aos detalhes internos, que podem trazer melhores resultados.

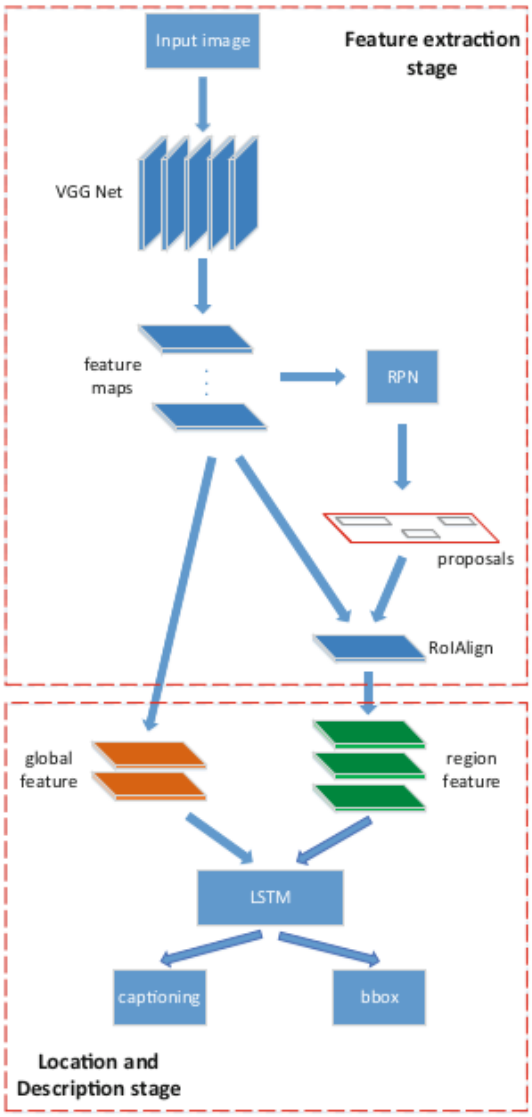
A arquitetura completa desenvolvida é ilustra na Figura 34, composta por duas etapas: extração de característica; localização e descrição. A primeira etapa, extração das características, é fortemente inspirada na Faster RCNN (Ren et al., 2017). A imagem completa é usada como entrada para a CNN, na forma da arquitetura VGG-16 (SIMONYAN; ZISSERMAN, 2015), que produz como resultado o mapa de característica. A partir das características, são obtidas as regiões, através da RPN, em que cada região é a entrada para a RoIAlign, sendo este o diferencial do trabalho de Zhang et al. (2019).

A RoIAlign foi desenvolvida para a arquitetura Mask RCNN (He et al., 2017), uma variação da Faster RCNN, em que não se utiliza a RoI comumente empregada. De modo geral, a RoIAlign transforma as características em um espaço maior, o que permite um melhor *"fit"* do *max polling*.

A etapa de localização e descrição é feita conforme a Figura 35, da mesma forma que o trabalho de Yang et al. (2017), ilustrado na Figura 31, inclusive utilizando a mesma função definida na Equação 4.13. Yang et al. (2017) utilizou uma LSTM para realizar a correção da *bbox*, outra para modelar o contexto (neste caso redefinido como características globais) e ainda outra LSTM para gerar a descrição, que combina as características globais com as características da região alvo. A diferença é que, no modelo de Yang et al. (2017), cada palavra gerada é utilizada também para fazer o ajuste da *bbox*, enquanto no trabalho de Zhang et al. (2019) utiliza-se a descrição final produzida.

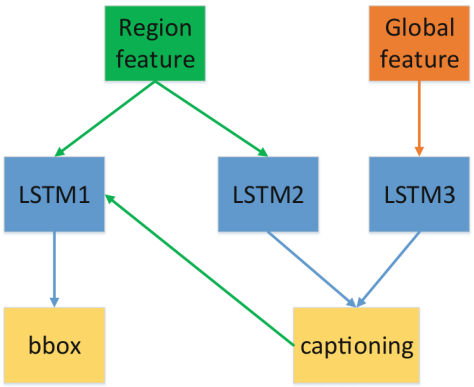
Os experimentos foram conduzidos com o *Visual Genome*, do qual foram selecionados apenas 94.313 imagens, destinando 5.000 para as partições de teste e validação (cada), e o restante para treinamento. Apenas as 10.000 palavras mais frequentes do vocabulário foram mantidas. O treinamento ocorreu durante 600.000 iterações, sendo a camada oculta da LSTM com tamanho de 512. O valor mAP alcançado por Zhang et al. (2019), 9,44, é maior que os de Johnson et al. (2016) e Yang et al. (2017), porém ainda inferiores aos 10,51 de Yin et al. (2019).

Figura 34 – Arquitetura proposta por Zhang et al. (2019)



Fonte: Retirado de Zhang et al. (2019)

Figura 35 – Arquitetura de geração das descrição de Zhang et al. (2019)



Fonte: Retirado de Zhang et al. (2019)

4.10 Considerações finais

Este capítulo apresentou os principais trabalhos de GDRI ou aplicações relacionadas, assim como suas técnicas e abordagens adotadas. Os trabalhos denotam forte utilização de técnicas de redes neurais artificiais para realização das tarefas, sobretudo da CNN para imagem e LSTM para o texto.

A Tabela 2 sumariza as principais informações de cada trabalho apresentado neste capítulo. A coluna de resultados exibe o melhor resultado reportado dentre todos os conjuntos de dados utilizados. Atualmente o modelo estado da arte para a tarefa de GDRI é o de Yin et al. (2019). Outros que destacam-se na área são os trabalhos de Johnson et al. (2016), Yang et al. (2017) e Zhang et al. (2019).

Tabela 2 – Resumo dos principais aspectos dos trabalhos relacionados

		Forma de Representação			Avaliação	
Trabalho	**Tarefa**	**Sentença**	**Imagem**	**Conjunto de dados**	**Métrica**	**Resultado**
Karpathy et al. (2014)	Recuperação de Informação	Relações de dependência	RCNN	Pascal1k, Flickr8K e Flickr30K	*Recall* e *Mean rank*	54,7 e 8
Karpathy e Fei-Fei (2015)	GDRI	RNN	RCNN	Flickr8K, Flickr30K e MSCOCO	BLEU-1	0,352
Zhang et al. (2015)	GDRI	LSTM	RCNN	Flickr8K e MSCOCO	BLEU-1	0,466
Johnson et al. (2016)	*Dense captioning*	LSTM	Variação da Faster-RCNN	Visual Genome 1.0	METEOR e mAP	0,273 e 5,39
Mao et al. (2016)	GER	LSTM	CNN	G-Refe UNC-Ref	Humana e *end-to-end*	20,4 e 0,833%
Yu et al. (2016)	GER	LSTM	CNN	G-Ref, RefCOCO e RefCOCO+	Humana, BLEU-2 e METEOR	76,14%, 0,277 e 0,151
Yang et al. (2017)	*Dense captioning*	LSTM[a]	Faster-RCNN	Visual Genome (VG) 1.0 e 1.2	mAP	9,31 (VG 1.0) e 9,96 (VG 1.2)
Yin et al. (2019)	*Dense captioning*	LSTM	Faster-RCNN	Visual Genome (VG) 1.0	METEOR e mAP	0,279 e 10,51
Zhang et al. (2019)	*Dense captioning*	LSTM[b]	Faster-RCNN	Visual Genome (VG) 1.0 e 1.2	mAP	9,44

Fonte: Elaborado pelo autor

[a] Também foram utilizadas outras duas LSTM, uma para modelar o contexto e outra a correção da *bbox*.
[b] Também foram utilizadas outras duas LSTM, uma para modelar o contexto e outra para os vizinhos próximos ao alvo.

Capítulo 5
MÉTODO GDRI-AMR

Este capítulo descreve o método desenvolvido neste projeto. Inicialmente, é detalhada a arquitetura da RNA apresentada por Johnson et al. (2016) (Seção 5.1), que foi usada como modelo para experimentação. Na Seção 5.2 são apresentadas as formas de representação experimentadas neste trabalho, sendo elas: língua natural, AMR, AMR anonimizada e AMR anonimizada concatenada. A Seção 5.3 apresenta como o conjunto de dados foi transformado nas respectivas formas de representação da sentença.

5.1 Arquitetura de Johnson et al. (2016)

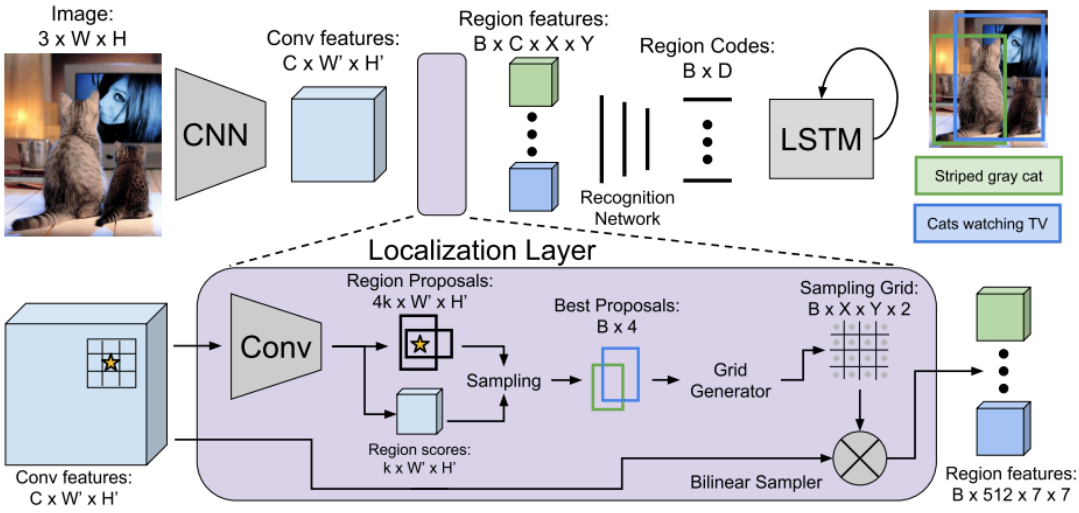
Este trabalho utilizou uma RNA com a arquitetura proposta por Johnson et al. (2016) e ilustrada na Figura 36, sendo esta uma variação da *Faster RCNN*. Optou-se por utilizar esta arquitetura devido ao fato de que, no início da realização deste trabalho, o método de Johnson et al. (2016) era o estado da arte na literatura para GDRI. O seu código fonte está disponível publicamente e o modelo original foi nomeado como *Fully Convolutional Localization Network* (FCLN).

A CNN foi aplicada com a arquitetura VGG-16, que consiste de 13 camadas de 3 x 3 intercaladas com 5 camadas de 2 x 2 de *max pooling*, removendo a última camada de *pooling* para que a partir de uma imagem de entrada no formato 3 x W x H, produza uma saída na forma C x W' x H', no qual $C = 512$, $W' = \frac{W}{16}$ e $H' = \frac{H}{16}$. A saída dessa rede é usada como entrada para rede de localização.

A principal diferença para a *Faster RCNN* é a substituição da RoI por um método de interpolação bilinear. A camada de localização seleciona B regiões de interesse e tem como saída três tensores de informação sobre as regiões, compostos por:

- Coordenadas das regiões: uma matriz B x 4 com as coordenadas de cada uma das *bbox*.
- Pontuação das regiões: um vetor de tamanho B com a pontuação de cada uma das regiões. Regiões com alta pontuação correspondem a regiões de maior interesse.

Figura 36 – Arquitetura do método GDRI



Fonte: Retirado de Johnson et al. (2016)

- Características das regiões: um tensor B x C x X x Y que possui características das regiões em uma grade X x Y de C-dimensões.

A regressão da *bbox* ocorre a partir de um centroíde (x_a, y_a), juntamente com largura (w_a) e altura (h_a), em que o modelo previu um escalar (t_x, t_y, t_w, t_h), de modo que a região de saída tenha como centro (x e y) e forma (w, h) dados por:

$$x = x_a + t_x w_a \tag{5.1}$$

$$y = y_a + t_y h_a \tag{5.2}$$

$$w = w_a exp(t_w) \tag{5.3}$$

$$h = h_a exp(h_w) \tag{5.4}$$

A RoI utilizada na *Faster RCNN* projeta cada região sobre um *grid* na forma W' x H' e é dividida em um *subgrid* X x Y, alinhando os pixels por arredondamento. A camada de RoI é uma função de duas entradas: as características convolucionais e as coordenadas das regiões propostas. Os gradientes podem ser propagados para as características, mas não para as regiões propostas.

Para contornar este problema, Johnson et al. (2016) utilizou a interpolação bilinear. A partir da entrada de um mapa da características U na forma C x W' x H' e a proposta de região,

ocorre a interpolação das características U para produzir a saída V na forma C x X x Y. Após fazer um *sampling* no *grid* G na forma X x Y x 2, associando cada elemento V com coordenadas reais sobre U. Se $G_{i,j} = (x_{i,j}, y_{i,j})$ então $V_{c,i,j}$ deve ser igual a U $(c, x_{i,j}, y_{i,j})$, desde que $(x_{i,j}, y_{i,j})$ sejam valores reais, então são transformados em um *kernel* de amostragem k, conforme a Equação 5.5, utilizando amostragem bilinear, correspondente ao kernel $k(d) = max(0, 1 - |d|)$.

$$V_{c,i,j} = \sum_{i'=1}^{W} \sum_{j'=1}^{H} U_{c,i',j'} k(i' - x_{i,j}) k(j' - y_{i,j}). \tag{5.5}$$

A amostragem do *grid* é uma função linear das regiões propostas, para que os gradientes possam ser retro propagados. A saída da camada de localização é a interpolação bilinear no formato B x C x X x Y, que é usada como entrada para a rede de localização.

A rede de localização é uma RNA totalmente conectada que processa as características das regiões provenientes da camada de localização. As características são comprimidas e transmitidas para duas camadas totalmente conectadas. Cada região produz um vetor de 4096 dimensões. Este vetor é usado como entrada para a geração da sentença.

Para geração da sentença em língua natural, foi empregada uma RNN do tipo LSTM. A partir da sequência de *tokens* $s_1, ...s_t$, a entrada da LSTM é $T + 2$ de vetores de palavras $x_{-1}, x_0x_1, ...x_T$, no qual $x_{-1} = CNN(I)$ é a região codificada e x_0 é o *token* especial de início (*START*).

Johnson et al. (2016) geram sentenças em LN diretamente. Dessa forma, sentenças que possuem palavras diferentes, mas expressam o mesmo significado, são aprendidas e preditas pelos modelos de forma independente. A representação AMR busca solucionar esse problema, representando os sentidos das sentenças de uma única forma.

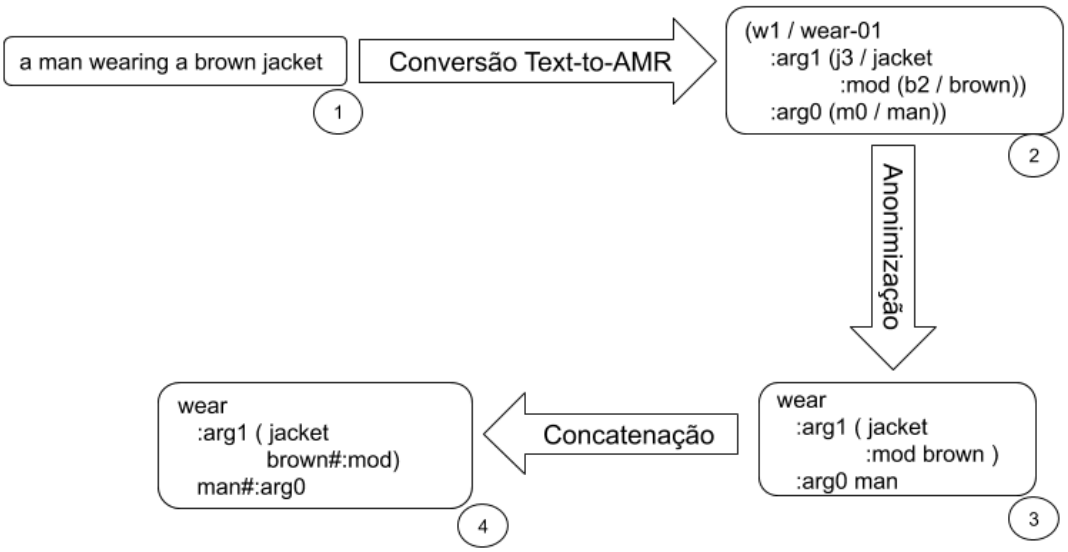
5.2 Formas de representação das sentenças

Com o objetivo de verificar se a representação semântica AMR é capaz de auxiliar na GDRI, diversas formas de representação das sentenças foram usadas como entrada para o código proposto por Johnson et al. (2016).

A Figura 37 ilustra as 4 formas de representação investigadas neste trabalho. A primeira representação (identificada pelo número 1 na figura) é a sentença em língua natural original disponível no conjunto de dados do *visual genome*, de agora em diante denominada **representação LN**. Essa é a forma de representação adotada por Johnson et al. (2016) e, portanto, usada para gerar o modelo *baseline* neste trabalho.

A segunda forma de representação é obtida através da transformação da representação LN para a representação semântica AMR através do parser *Text-to-AMR* de Lyu e Titov (2018).

Figura 37 – Formas de representação das sentenças



Fonte: Elaborado pelo autor

Neste trabalho, esta representação é denominada **representação AMR**, sendo considerada sua forma linearizada.

A AMR pode ser simplificada em uma espécie de anonimização, conforme especificado no trabalho de Konstas et al. (2017). A anonimização tem como finalidade diminuir substancialmente a complexidade dos grafos AMR linearizados e propiciar uma melhor aprendizagem para os modelos, especialmente os que utilizam RNA. Para isso, busca diminuir a esparsidade das palavras na AMR, como entidades nomeadas e quantificadores.

A simplificação do grafo é realizada retirando nomes de variáveis e barras oblíquas ("/") antes de conceitos. No caso da variável ser mencionada em outros lugares no grafo, a mesma é substituída pelo seu conceito. Também é removida a identificação dos sentidos dos verbos AMR (identificados pelo caractere "-" seguido de um número, logo após o verbo).

A anonimização de entidades nomeadas ocorreu para todas as informações que continham a tag *:name*, o que inclui pessoas, países e também outras diversas entidades marcadas com **-enitity* e quantificadores (**-quantity*). Da mesma forma, para redução da esparsidade, cada entidade nomeada é substituída por um dos 4 tipos do Stanford *Named Entity Recognizer* (NER): pessoa, localização, entidade e diverso. Esse processo de anonimização é totalmente automático e reversível, sendo possível reconstruir a AMR original a partir de sua versão anonimizada. Assim, a terceira forma de representação das sentenças foi obtida através do processo de anonimização. A partir das sentenças na forma de AMR linearizada, as mesmas são anonimizadas. Na sequência deste trabalho essa representação será referenciada como **AMR anonimizada** (AMR Anon.).

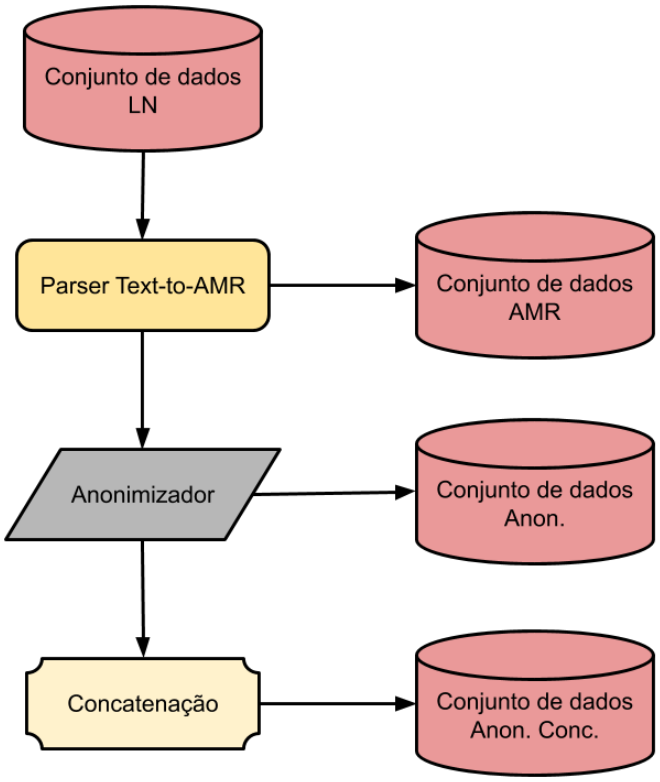
A quarta forma de representação da sentença foi obtida realizando a concatenação da representação semântica AMR anonimizada. Essa forma de representação foi gerada na tentativa

de manter uma relação mais forte entre o argumento do verbo (ou atributo de um conceito) e a sua palavra imediatamente posterior, ou seja, a palavra associada ao argumento (ou ao conceito). Para isso, a concatenação foi realizada se a *tag* que representa algum argumento, iniciada com dois pontos (“:”), é sucedida por um *token* que não seja um abre parênteses (“(”). Como caractere delimitador dessa concatenação, inserido para facilitar posteriormente a separação dos *tokens*, utilizou-se o cerquilha (“#”). Essa representação foi denominada **AMR anonimizada concatenada** (AMR Anon. Conc.).

5.3 Transformação do conjunto de dados

A partir do conjunto de dados do *visual genome* (que será detalhado na Seção 6.1), todas as sentenças são transformadas nas outras 3 formas de representação, obtendo-se assim o mesmo conjunto original (representação LN), porém com formas diferentes de representação das mesmas sentenças. A geração desses 4 conjuntos de dados é apresentada na Figura 38. A imagem também possui uma outra representação, AMR anonimizada concatenada 2 (anon. conc. 2), na qual é as sentenças estão no mesmo formato que a representação AMR anonimizada concatenada, mas buscando valores próximos em relação a sua parametrização (que será detalhada na Seção 6.3.1).

Figura 38 – Geração dos conjuntos de dados

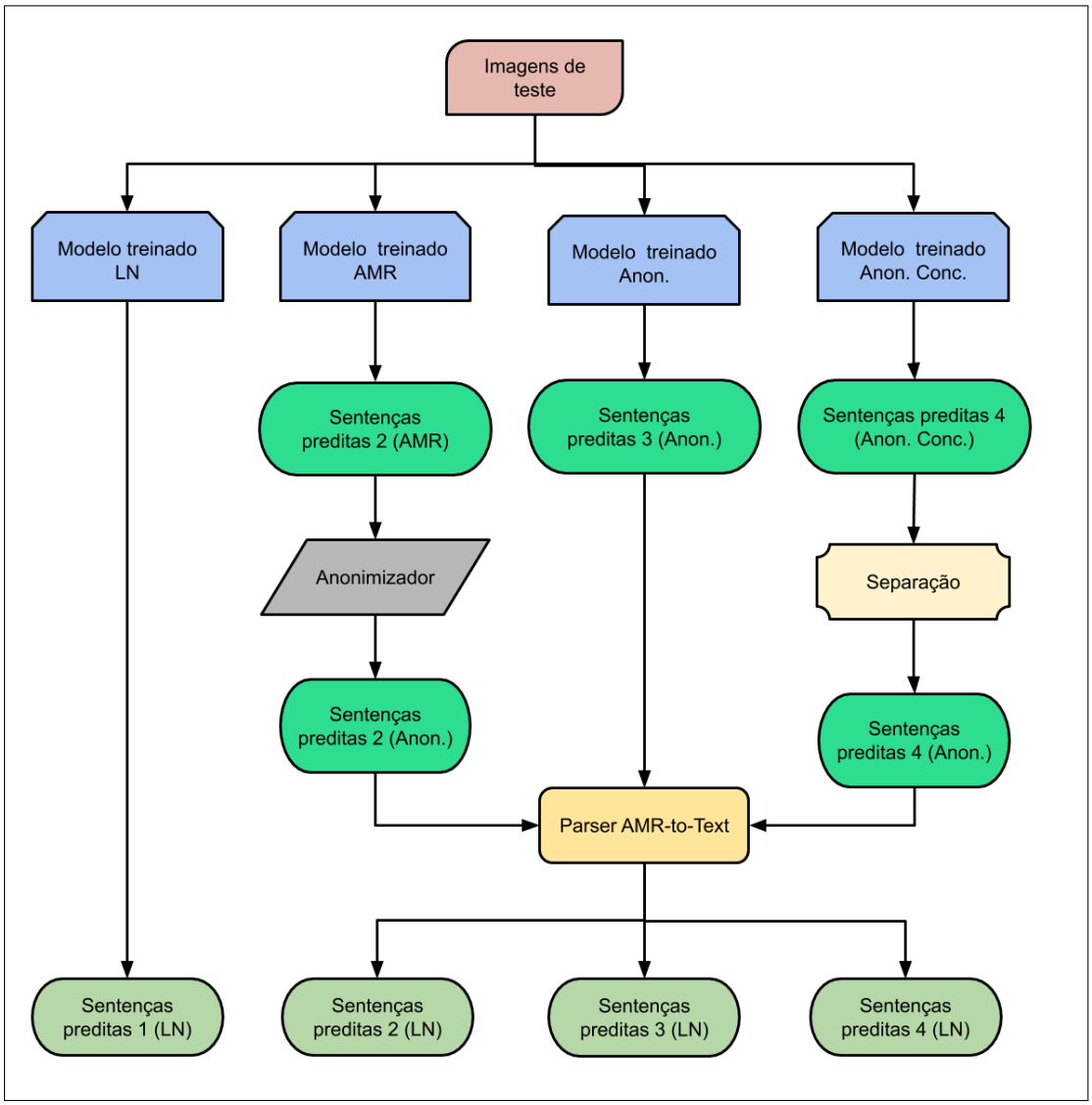


Fonte: Elaborado pelo autor

A Figura 39 ilustra o processo de predição das descrições das regiões das imagens

presentes no conjunto de teste. Assim, por exemplo, a partir da imagem apresentada na Figura 40, as sentenças que foram geradas por cada um dos modelos são apresentadas na Tabela 3, na qual a coluna "sentença predita" é a sentença da forma como foi predita pelo modelo, e a coluna "sentença transformada em LN" representa a sentença predita pelos modelos após serem transformadas em LN.

Figura 39 – Processo de predição dos modelos de GDRI



Fonte: Elaborado pelo autor

Como pode ser visto pelas sentenças geradas pelos modelos (apresentadas na Tabela 3), a sentença em língua natural faz referência a um carro estar ao lado do outro. A sentença predita em AMR, por sua vez, fala sobre a roda de um ônibus que não aparece na região da imagem, ou seja, uma descrição errada dado o contexto. Já as sentenças geradas pelos modelos que usam AMR Anonimizada, tanto na sua forma linearizada, quanto na concatenada, produziram descrições possíveis para a região da imagem: uma descrevendo as janelas dos carros e outra descrevendo o carro preto estacionado.

Figura 40 – Exemplo do conjunto de dados

Fonte: ***Dataset Visual Genome***

Tabela 3 – Sentenças preditas pelos modelos investigados neste trabalho para a imagem da Figura 40

	Sentença predita	**Sentença transformada em LN**
LN	*a car on the side of a car*	–
AMR	(w0 / wheel :location (b1 / bus))	*the wheel on the bus*
AMR Anon.	window :part-of car	*windows of cars*
AMR Anon. Conc.	park :arg1 (car black#:arg1-of)	*the black car is parked*

Fonte: Elaborado pelo autor

Como é possível notar pelas sentenças preditas pelos modelos que usaram AMR, as relações entre conceitos/argumentos são preservadas na sentença predita e podem gerar descrições mais ricas em língua natural. Esta é a hipótese perseguida neste trabalho e que foi avaliada nos experimentos descritos no próximo Capítulo.

Capítulo 6
EXPERIMENTOS E RESULTADOS

Para avaliar métodos de GDRI de maneira automática, são necessários três componentes: (1) um conjunto de dados de imagens, com regiões e descrições, (2) um modelo que seja capaz de gerar descrições para regiões da imagem e (3) uma forma de avaliar a qualidade do modelo.

Quanto ao primeiro item, o conjunto de dados usado nos experimentos deste trabalho é descrito na Seção 6.1. O modelo-base para geração de descrições de regiões da imagem adotado neste trabalho foi descrito na Seção 5.1. Para facilitar a reprodução dos experimentos, também são descritos: o pré-processamento para preparação dos dados (Seção 6.2) e o treinamento realizado para os vários modelos experimentados (Seção 6.3). Por fim, utilizando as medidas de avaliação apresentadas na Seção 3.4, a Seção 6.4 descreve a metodologia de avaliação e os resultados obtidos por cada um dos modelos avaliados.

6.1 Conjunto de dados

Os modelos de aprendizado de máquina necessitam aprender a realizar a tarefa para qual foram planejados. Esse processo de aprendizado geralmente é realizado a partir de exemplos, ou seja, exemplos reais daquilo que o modelo deve aprender. A partir desses exemplos, o modelo deve ser capaz de melhorar a sua capacidade de desempenhar a tarefa planejada à medida que novos exemplos são apresentados. Ao conjunto de vários exemplos é dado o nome de conjunto de dados.

O conjunto de dados (do inglês *dataset*) utilizado neste trabalho é o Visual Genome[1] (VG) proposto por Krishna et al. (2017). O VG foi criado com o objetivo de prover um conjunto de dados grande e representativo para tarefas multimodais (processamento de imagem e texto), como *visual answers*, *questions answers*, legendas de imagem, descrições de regiões de imagem, entre outros.

Cada imagem do VG é acompanhada por: descrições de regiões que correspondem a parte da imagem; e dois pares de *question answer*, sendo uma pergunta e resposta em forma de

[1] Disponível em http://visualgenome.org/

texto, e outra pergunta em texto e resposta na forma de uma *bbox*. Como o foco deste trabalho é na GDRI, serão descritos os passos utilizados para construir o conjunto de descrições para as regiões de imagem. A versão 1.0 do VG, que foi a utilizada neste trabalho, assim como em Johnson et al. (2016), foi criada a partir da intersecção dos conjuntos *Yahoo Flickr Creative Commons 100 Million* (YFCC100M) (THOMEE et al., 2016) e *Microsoft Common Objects in Context* (MSCOCO) (LIN et al., 2014).

O conjunto completo possui 108.077 imagens. As regiões de uma imagem podem ter sobreposições, desde que suas respectivas sentenças sejam diferentes. Cada imagem possui uma média de 50 regiões, e cada descrição contém no máximo 16 palavras.

O VG foi coletado e verificado inteiramente através da plataforma de *crowdsourcing* *Amazon Mechanical Turk*[2] (AMT). Nesse processo, para cada nova imagem a ser anotada, o anotador marcava 3 *bbox* e fornecia uma descrição para cada. Para evitar que descrições semelhantes fossem geradas, para cada nova descrição foi calculado o valor de BLEU (PAPINENI et al., 2002) entre essa nova descrição e todas as sentenças da imagem e as 100 descrições mais próximas entre si entre todas as imagens. Caso o valor de BLEU fosse maior do que 0,7 para algum par, a nova descrição não foi aceita.

Outro requisito utilizado para aceitar uma descrição foi que todos os objetos mencionados na descrição estivessem dentro da área da *bbox*. Essa verificação foi possível pois os conjuntos de dados utilizados para criação do VG já possuíam os objetos segmentados e classificados, ambas as tarefas realizadas manualmente.

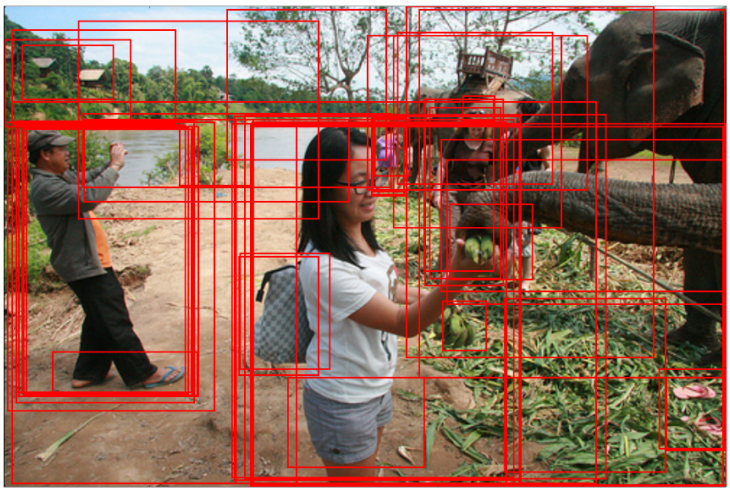
Uma última verificação foi realizada fornecendo a *bbox* e a respectiva descrição a 3 pessoas no AMT para que assinalassem se estavam corretas. Foram consideradas válidas as descrições assinaladas como corretas por pelo menos duas pessoas.

Esse processo de delimitação e descrição das regiões ocorreu iterativamente, até que cada imagem tivesse cerca de 50 regiões válidas. Um exemplo de uma imagem do VG e suas respectivas regiões é ilustrado na Figura 41, e algumas descrições dessa mesma imagem são apresentadas na Figura 42.

Os autores também realizaram uma experimentação para verificar a diversidade semântica das descrições fornecidas. Para isso, utilizaram o modelo de *embedding word2vec* (MIKOLOV et al., 2013), pré treinado no Google News[3]. Cada palavra das descrições foi convertida em um vetor de 300 dimensões. A descrição completa foi representada como a média dos vetores das palavras, desconsiderando as *stop words*, conforme ilustrado na Figura 43. Um agrupamento hierárquico aglomerativo (STEINBACH et al., 2000) foi utilizado nos vetores das descrições, e foram encontrados 71 agrupamentos, dos quais alguns podem ser vistos na Figura 44. Em média, cada imagem contém descrição de 17 diferentes agrupamentos. O menor número agrupamentos encontrados em uma imagem foi 4 e o maior, 26.

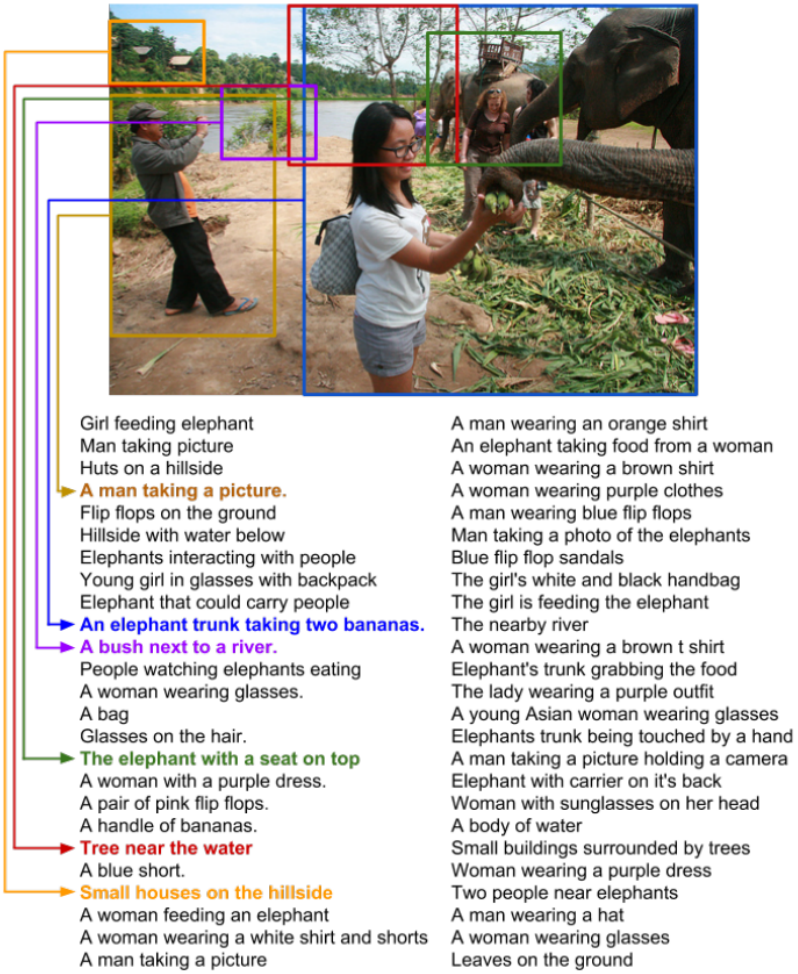
[2] Disponível em www.mturk.com/
[3] Disponível em www.news.google.com

Figura 41 – Exemplo de imagem e regiões do *Visual Genome*



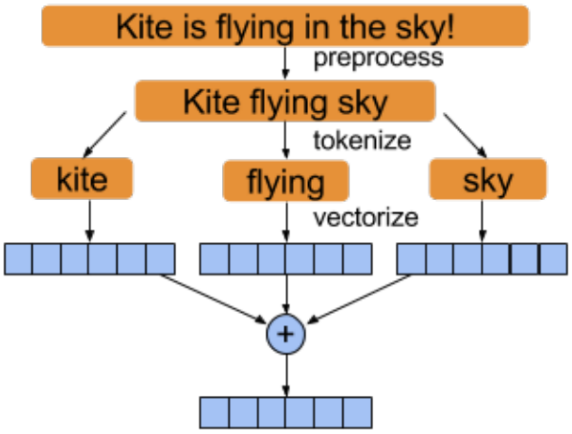
Fonte: Retirado de Krishna et al. (2017)

Figura 42 – Exemplo de regiões e suas respectivas descrições do *Visual Genome*



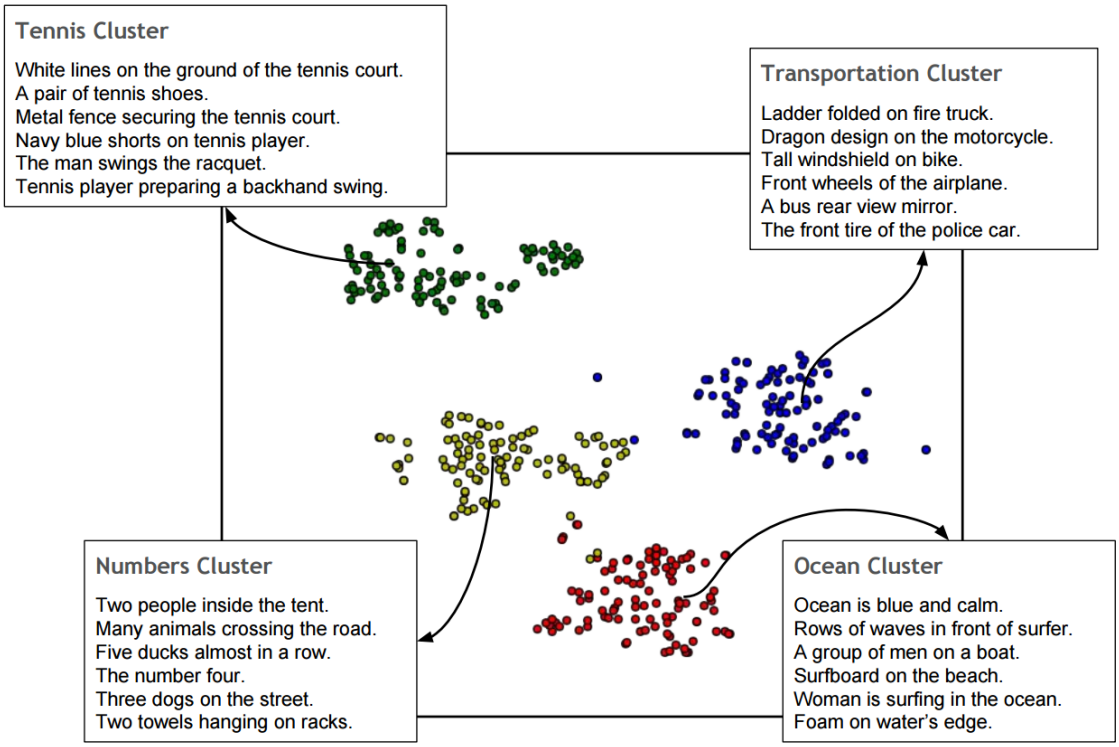
Fonte: Retirado de Krishna et al. (2017)

Figura 43 – Descrição do *Visual Genome* sendo transformada em uma representação vetorial distribuída



Fonte: Retirado de Krishna et al. (2017)

Figura 44 – Exemplo de agrupamento semântico do *Visual Genome*



Fonte: Retirado de Krishna et al. (2017)

6.2 Pré-processamento

Os mesmos passos de pré-processamento e parâmetros de Johnson et al. (2016) foram utilizados neste trabalho para gerar diferentes modelos com base nas várias formas de representação da sentença, conforme detalhado na Seção 5.2.

Primeiramente, a partir do conjunto original de 108.077 imagens foram retiradas as que

continham menos de 20 ou mais que 50 regiões, para reduzir a variação no número de regiões por imagem. Essa etapa, apesar de descrita no artigo de Johnson et al. (2016), não estava disponível no seu código fonte. Esse processamento resultou em um total de 4.191.170 regiões/sentenças (uma sentença para cada região), reduzindo o conjunto efetivamente usado nos experimentos para 91.610 imagens. No entanto, Johnson et al. (2016) reportam que esse processo resultou em um valor diferente de imagens (87.398 imagens) do que realizado neste trabalho.

6.2.1 Parâmetros

Seguindo os parâmetros estabelecidos por Johnson et al. (2016) e o seu próprio código fonte disponível, regiões cuja a sentença de descrição continha mais de 10 *tokens* foram descartadas e *tokens* que ocorriam menos de 15 vezes no conjunto total foram trocados por <*UNK*>. Imagens para as quais todas as regiões foram descartadas, também foram excluídas. Este processo foi realizado para todas as formas de representação da sentença.

Na Tabela 4 é possível encontrar todos os parâmetros de pré-processamento aplicados em cada uma das formas de representação das sentenças, assim como o tamanho do conjunto resultante em termos da quantidade de imagens, sentenças e tamanho do vocabulário, no qual o valor antes da barra indica o total de *tokens* únicos (*types*) e o valor após a barra, a quantidade total de ocorrências desses *tokens*. Nota-se que o conjunto de dados resultante foi diferente para diversas formas de representação, visto que o processo de transformação entre as formas de representação da sentença pode acarretar em um número maior ou menor de *tokens*, e consequentemente, impactar na frequência de ocorrência dos mesmos.

A Tabela 4 possui também uma outra AMR anonimizada concatenada, nomeada como **AMR anonimizada concatenada 2**. Essa representação segue a mesma ideia da AMR anonimizada concatenada, só que agora com o limite máximo para tamanho da sentença aumentado para 20 de modo a considerar, no conjunto de dados, uma quantidade de sentenças mais próxima ao *baseline* em língua natural (LN).

Para relembrar, a partir das formas de representação da sentença (descritos na Seção 5.2), os modelos experimentos podem ser descritos, de maneira simplificada, como:

1. **LN** (*baseline*) – treinado com conjunto de dados em língua natural (LN);
2. **AMR** – treinado com o conjunto de dados AMR;
3. **Anon.** – treinado com o conjunto de dados AMR anonimizado;
4. **Anon. Conc.** – treinado com o conjunto de dados AMR anonimizado e concatenado;
5. **Anon. Conc. 2** – treinado com o conjunto de dados AMR anonimizado e concatenado para manter uma quantidade próxima das sentenças ao *baseline*.

Tabela 4 – Parametrização do pré-processamento

Representação	Parâmetros		Conjunto Resultante		
	Freq. Mín.	Tam. Máx.	Imagens	Sentenças	*Types*/Nº Ocorr.
LN	15	10	91.610	4.168.246	12.514/20.915.384
AMR	15	10	90.193	989.079	4.782/6.752.935
AMR anon.	15	10	91.607	3.135.352	6.038/18.275.781
AMR anon. conc.	15	10	91.609	3.548.212	18.601/17.173.659
AMR anon. conc. 2	15	20	91.609	4.165.814	21.928/25.106.949

Fonte: Elaborado pelo autor

6.2.2 Particionamento

Como forma de particionar, ou seja, forma de dividir os exemplos em treinamento, validação e teste, seguiu-se o estabelecido por Johnson et al. (2016), em que do total de exemplos utilizados no artigo original (87.398), utilizou-se 5.000 para validação, 5.000 para teste (correspondente a aproximadamente 5,72% para cada partição) e o restante para treinamento (77.398).

Neste trabalho, o conjunto de dados foi particionado da seguinte forma: 5,45% para validação (5.001 exemplos); 5,45% para teste (5.001 exemplos); e o restante dos exemplos para treinamento (81.608).

6.2.3 Codificação da sentença

A codificação[4] realizada por Johnson et al. (2016) ocorreu atribuindo um número, de forma sequencial, para cada palavra do vocabulário. Posteriormente, os *tokens* de cada uma das sentenças são substituídos pelo respectivo número, cujo valor foi associado ao *token* no vocabulário. Essa forma de codificação foi utilizada no treinamento de todos os modelos.

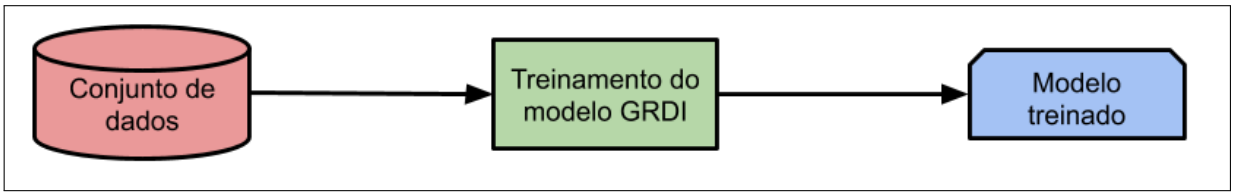
6.3 Treinamento

Os modelos foram treinados utilizando o código de Johnson et al. (2016) variando-se apenas a entrada para gerar um modelo para cada uma das formas de representação das sentenças e combinações de parâmetros sumarizados na Tabela 4. Assim, ao todo cinco modelos foram treinados, um para cada uma das formas de representação e parâmetros definidos nesta tabela.

Conforme ilustrado na Figura 45, o treinamento tem como entrada um dos 5 conjuntos de dados produzidos como descrito anteriormente (veja Figura 38 e Tabela 4). O mesmo particionamento de treinamento, validação e teste (veja Seção 6.2.2) foi usado para todos os modelos, sendo os exemplos nas partições de treinamento e validação usados para gerar os modelos e os de teste aplicados na avaliação.

[4] Neste trabalho definida como a transformação da palavra em um vetor numérico.

Figura 45 – Processo de treinamento GDRI



Fonte: Elaborado pelo autor

6.3.1 Parâmetros

Os modelos possuem uma grande quantidade de parâmetros que podem ser modificados (aproximadamente 60). Como não existe um detalhamento e especificação dos valores para cada um dos parâmetros no trabalho de Johnson et al. (2016), foram usados os valores padrões contidos no código fonte.

Dentre esse parâmetros, destacam-se o número de neurônios de cada camada da LSTM (512), o tamanho do vetor da palavra predita pela LSTM (512), dropout (0.5) e taxa de aprendizado (1e-6). Enquanto para a RPN, número de neurônios da camada extra (512), peso de 0.05 para a regressão da *bbox* e peso da classificação (0.1).

No artigo de Johnson et al. (2016), não foi estipulado qual critério foi empregado para interromper o treinamento, apenas a duração do mesmo (cerca de 5 dias). Como essa duração pode variar de acordo com a configuração da máquina, optou-se por utilizar o número de iterações do modelo como critério de parada. Assim, cada modelo foi treinado por 1 milhão de iterações, com um *checkpoint* armazenado a cada cem mil iterações para salvar os pesos do modelo, totalizando 10 *checkpoints* para cada modelo treinado. O processo de treinamento, para cada um dos modelos, levou aproximadamente 7 dias para ser concluído. A máquina utilizada no treinamento conta com o sistema operacional Ubuntu 16.04, equipado com 2 processadores Intel Xeon E5-2683, com 80 MB de memória cache e 64 núcleos de processamento e uma memória de 252 GB. Como *Graphics Processing Unit* (GPU), foi usada uma NVIDIA Tesla P100 com 16GB.

6.4 Resultados

Após o treinamento dos 5 modelos investigados neste trabalho, procedeu-se com a etapa de teste e avaliação. Os modelos de 1 a 4 correspondem a cada uma das 4 formas de representação (vide Seção 5.2) e o modelo 5 corresponde ao melhor modelo (4) mantendo uma maior proximidade de quantidade de sentenças que o modelo de *baseline*.

A avaliação das sentenças preditas para o conjunto de teste foi realizada de duas maneiras, como explicado nas próximas subseções: avaliação automática de todo o conjunto de teste (cerca de 5.000 exemplos) usando as medidas mAP, SMATCH, BLEU 1 e METEOR, e por meio de

uma análise manual de cerca de 350 exemplos selecionados aleatoriamente.

6.4.1 Avaliação automática das sentenças preditas

Como pode ser observado no Capítulo 4 e sumarizado na Tabela 2, a mAP é atualmente a principal medida utilizada para comparação dos modelos de GDRI. A vantagem na sua utilização em relação a BLEU e METEOR é que, apesar de fazer uso desta última para considerar uma sentença correta, utiliza vários limiares de IoU e de similaridade entre as sentenças (mais detalhes na Seção 3.4.2.1).

A avaliação com a mAP ocorreu exatamente sobre as sentenças preditas pelos modelos em relação a suas respectivas referências. Dessa forma, o modelo 1 (*baseline*) foi o único em que a avaliação ocorreu em LN. Para os demais modelos, de 2 a 5, a avaliação ocorreu sobre suas sentenças preditas, sendo AMR para o modelo 2, AMR anonimizada para o modelo 3 e AMR anonimizada concatenada para os modelos 4 e 5. Para que essa avaliação fosse possível, as sentenças de referência (*gold standard*) usadas no cálculo da mAP também passaram pelo mesmo processamento das sentenças sendo avaliadas, ou seja, foram convertidas para AMR, anonimizadas e concatenadas, conforme a necessidade.

Os resultados de mAP para os modelos são exibidos na Tabela 5. Observa-se que os valores obtidos pelos modelos propostos neste trabalho (2 a 5) foram melhores que os obtidos pelo *baseline*. No artigo de Johnson et al. (2016), os autores reportam um valor de 5,39 para o modelo treinado em LN. No experimento realizado com os mesmos valores de parâmetros disponíveis, foi obtido um valor de 4,82.

Tabela 5 – Resultados da avaliação sobre as sentenças preditas pelos modelos com mAP

Modelo	mAP
1	4,824847
2	4,833836
3	7,016648
4	8,617150
5	**8,978977**

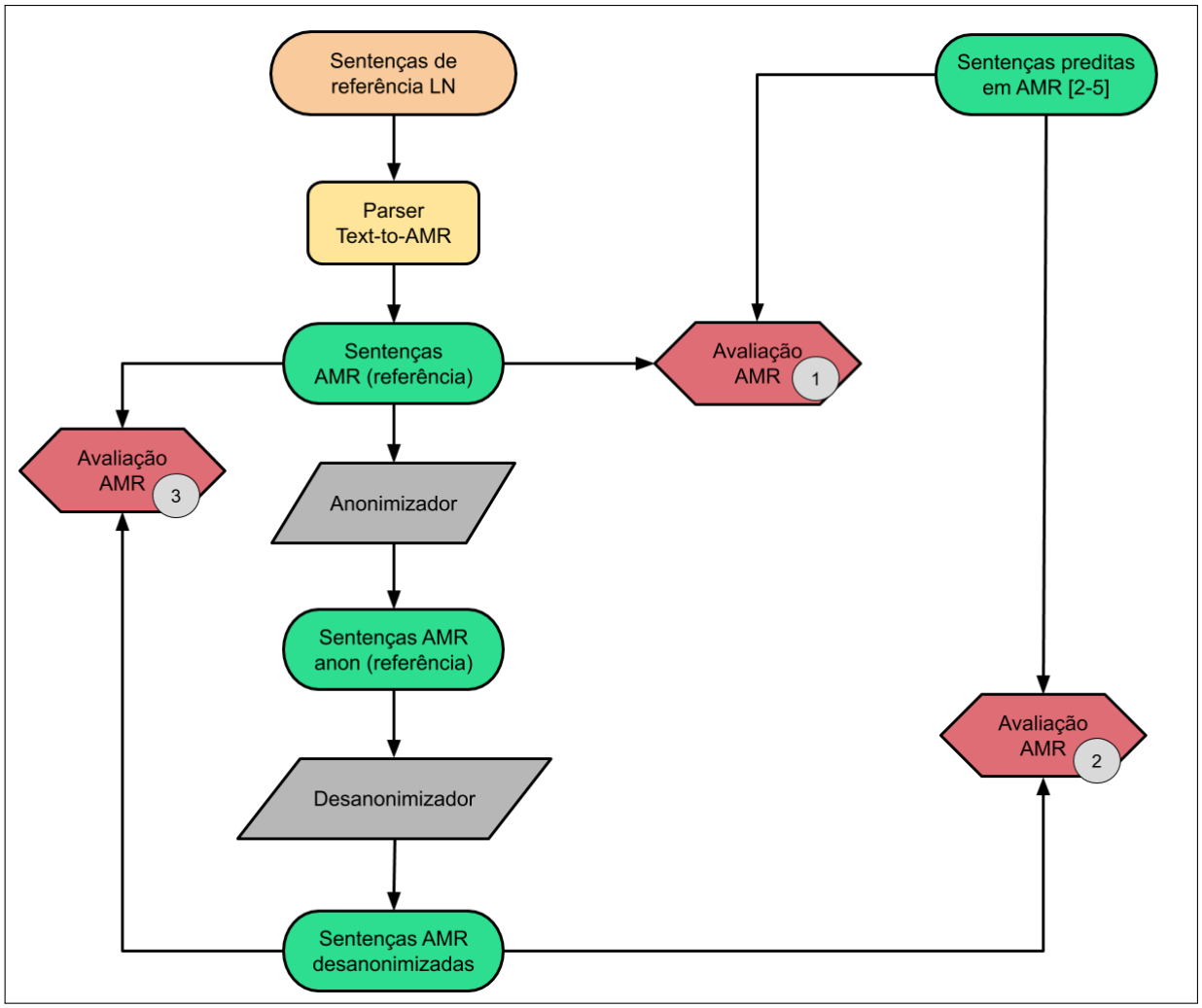
Fonte: Elaborado pelo autor

Entre os modelos propostos neste trabalho, o modelo 2 foi o que obteve o menor valor de mAP (4,83), enquanto os modelos 4 e 5 obtiveram os melhores resultados, 8,61 e 8,97, respectivamente, representando um ganho de aproximadamente 80% em relação ao *baseline* experimentado. Dessa forma, podemos concluir que a AMR em sua forma anonimizada concatenada se mostra efetiva, em termos de mAP, para a GDRI.

6.4.2 Avaliação automática das sentenças em AMR

Os modelos de 2 a 5 predizem alguma forma de representação AMR. Dessa forma, antes de serem transformados em LN, também foram avaliados de acordo com a medida apropriada para a representação semântica, a SMATCH, conforme explicado na Seção 3.4.2.2. Contudo, o processo de conversão para AMR (e posterior anonimização e desanonimização), por ser realizado por meio de métodos automáticos, insere ruídos e erros na predição. Com o intuito de verificar o impacto desses processos, uma forma de avaliação foi proposta. A Figura 46 ilustra o processo de avaliação utilizado para a avaliação das sentenças AMR.

Figura 46 – Processo de avaliação AMR



Fonte: Elaborado pelo autor

Primeiramente, a SMATCH foi calculada considerando-se a sentença AMR predita comparada à sentença de referência (*gold standard*) convertida para AMR (avaliação 1). O intuito dessa avaliação era calcular a SMATCH desconsiderando-se qualquer perda ocasionada pelos processos de anonimização e desanonimização. Em outras palavras, essa avaliação ocorre entre as predições ideais, que o modelo deveria predizer, e as que ele realmente predisse. Em

seguida, realizou-se a avaliação considerando a perda ocasionada por esses processos (avaliação 2). Por fim, avaliou-se a perda do processo de conversão AMR e anonimização comparando-se a sentença de referência com ela mesma, mas após a conversão para AMR, anonimização e desanonimização (avaliação 3).

A Tabela 6 apresenta os resultados obtidos por cada um dos modelos em seus respectivos pares de avaliação, em que as avaliações 1, 2 e 3 na Figura 46 correspondem às avaliações "referência e predita", "desanonimizada e predita", e "referência e desanonimizada", respectivamente. Os valores da coluna "referência e desanonimizada" possuem uma pequena diferença pois como a arquitetura completa é treinada, ou seja, tanto a detecção das *bbox* quanto a geração das sentenças propriamente dita, pode resultar em um número diferente de *bbox* para a mesma imagem em cada um dos modelos. Contudo, ressalta-se que foram utilizados os mesmos exemplos de testes em todos os modelos.

As **sentenças de referência** são as referências em LN, transformadas em AMR através do *parser Text-to-AMR*. As **sentenças desanonimizadas** são as referências em AMR que foram anonimizadas, e novamente transformadas em AMR. As **sentenças preditas**, como o próprio nome sugere, são as sentenças preditas pelos modelos de 2 a 5.

Tabela 6 – Resultados da avaliação AMR com SMATCH

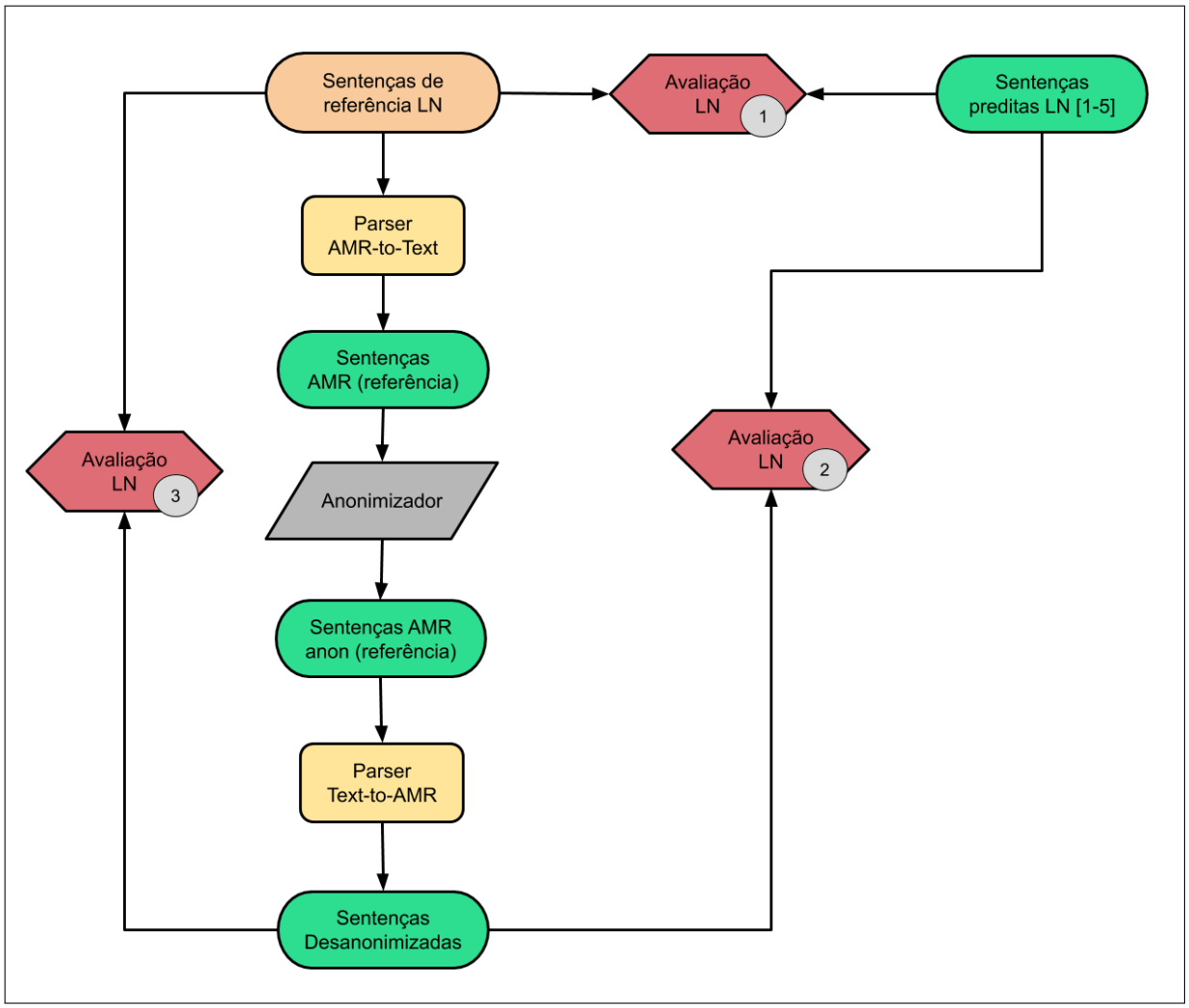
Modelo	Referência e predita	Desanonimizada e predita	Referência e desanonimizada
1	-	-	-
2	0,2021	0,1857	**0,8144**
3	**0,2058**	0,2432	0,7759
4	0,2039	**0,2472**	0,7748
5	0,1919	0,2337	0,7647

Fonte: Elaborado pelo autor

Os resultados demonstram que na conversão de LN para AMR um erro significativo de cerca de 20% (avaliação referência e desanonimizada), uma vez que comparando a sentença de referência "com ela mesma" após o processo de desanonimização, deveria resultar em um SMATCH de 1,0 e não de apenas 0,76-0,81. A avaliação entre as sentenças desanonimizadas e preditas foram um pouco melhores que as referências e preditas. Esse melhor resultado justifica-se pelo fato de que no primeiro par de avaliação, ambas as sentenças passaram pelos *parsers* e anonimizadores, e consequentemente, também possuem ruídos de tais ferramentas.

Em relação aos baixos valores obtidos para a medida SMATCH, vale ressaltar que embora ela leve em consideração a estrutura do grafo AMR e, desse modo, traga consigo uma flexibilidade maior do que as medidas que comparam língua natural (como a METEOR), ela não é a mais indicada para avaliação de GDRI. Isso porque é possível gerar uma descrição correta para uma dada *bbox* que seja completamente diferente da descrição de referência em termos dos conceitos utilizados e relações entre eles. Assim, a SMATCH foi apresentada neste trabalho

Figura 47 – Processo de avaliação LN



Fonte: Elaborado pelo autor

apenas para complementar a avaliação específica da AMR, mas não como um indicativo da efetividade do modelo para GDRI.

6.4.3 Avaliação automática das sentenças em língua natural

A avaliação automática da qualidade das sentenças preditas pelo modelo é parte essencial para confirmação ou refutação da hipótese deste trabalho. Assim, conforme detalhado na Seção 3.4, existem algumas medidas que podem ser utilizadas para contrastar uma sentença em língua natural em relação a outra. Neste trabalho, as sentenças em LN preditas pelos 5 modelos foram comparadas com as sentenças de referência presentes no conjunto de teste por meio de três maneiras, descritas na Figura 47.

A primeira avaliação, **sentenças de referência e preditas** (avaliação 1), tem como finalidade avaliar o processo completo, na forma *end-to-end*. Para isso, compara as sentenças dos exemplos de teste em LN (*gold standard*) e as sentenças preditas pelo modelo. Em outras

palavras, essa avaliação ocorre entre as sentenças ideais, que o modelo deveria predizer, e as sentenças que o modelo predisse. Quanto maior o valor das medidas nesse caso, mais próxima a sentença predita está da sentença de referência em termos de estrutura sintática.

A segunda avaliação, **sentenças desanonimizadas e preditas** (avaliação 2), avalia as sentenças preditas pelo modelo em contraponto aos erros que os *parsers Text-to-AMR* e *AMR-to-Text* produziram. Para isso, compara as sentenças preditas pelo modelo com as sentenças dos exemplos de teste anonimizados que foram transformadas em LN pelo *parser AMR-to-Text*. Dessa maneira, ao aplicar o mesmo processo de conversão texto-AMR-texto nas sentenças de referência e predita, espera-se ter uma noção da perda que essas conversões acarretam.

A terceira avaliação, **sentenças de referência e desanonimizadas** (avaliação 3), tem como objetivo avaliar a referência em LN (*gold standard*) em relação aos erros gerados pelos *parsers Text-to-AMR* e *AMR-to-Text*. Para tal, compara as sentenças de referência antes e depois da conversão texto-AMR-texto. Dessa forma, essa avaliação tem como propósito identificar exatamente a perda acarretada pelo processo de conversão texto-AMR-texto.

As medidas usadas para avaliação das sentenças em LN foram: a BLEU-1 e METEOR. Ambas as medidas medem a semelhança das palavras presentes entre as sentenças de referência e predita. Tais métricas foram detalhadas nas Seções 3.4.1.1 e 3.4.2.2 respectivamente. Os resultados para essas medidas, em cada um dos processos de avaliação ilustrados na Figura 47, são apresentados nas próximas subseções.

6.4.3.1 Resultados da Avaliação 1 – Sentenças de referência e preditas

A avaliação entre as sentenças de referência e predita tem como finalidade avaliar o processo completo (*end-to-end*) dos modelos. Os resultados obtidos para todos os modelos são exibidos na Tabela 7.

Tabela 7 – Resultados da avaliação 1: sentenças de referência e preditas

Modelo	BLEU 1	METEOR
1	**0,217538**	**0,111240**
2	0,168880	0,087836
3	0,159005	0,089734
4	0,155306	0,087802
5	0,149260	0,084143

Fonte: Elaborado pelo autor

Observou-se que o modelo treinado com LN obteve um resultado melhor em comparação aos demais modelos nas medidas BLEU e METEOR. O modelo de LN obteve na BLEU-1 um resultado de 0,217 contra 0,168 do modelo AMR (melhor modelo). Contudo, como já mencionado, essa maior proximidade entre a sentença predita e a sentença de referência, em

termos de palavras e n-gramas, não necessariamente reflete a qualidade da descrição gerada no caso de GDRI.

Apesar do modelo 5 ter mostrado-se superior ao modelo 4 quando avaliado com a medida mAP, essa melhora não refletiu-se na avaliação entre as sentenças de referência e preditas em LN, obtendo o pior valor em BLEU-1. Na medida METEOR, o modelo 2 obteve o maior valor, seguido pelo modelo 4.

6.4.3.2 Resultados da Avaliação 2 – Sentenças desanonimizadas e preditas

A avaliação entre as sentenças desanonimizadas e preditas teve como objetivo medir o erro que o *parser AMR-to-Text* produz. Como o *parser* não foi treinado no conjunto de dados do VG e por também utilizar técnicas de RNA, um ruído (erro) é esperado. Esse erro é válido tanto para conversão das sentenças preditas pelos modelos de 2 a 5 e transformadas em LN, quanto para as sentenças desanonimizadas a partir das sentenças de referência. Os valores obtidos são exibidos na Tabela 8.

Tabela 8 – Resultados da avaliação 2: desanonimizadas e preditas

Modelo	BLEU 1	METEOR
1	0,176834	0,094895
2	0,180773	**0,115417**
3	0,181590	0,107765
4	**0,182515**	0,108703
5	0,171325	0,100587

Fonte: Elaborado pelo autor

Considerando-se os valores obtidos, observou-se que os modelos de 2 a 4 foram superiores ao modelo 1 (*baseline*) em ambas as medidas (BLEU 1 e METEOR), corroborando que a transformação de LN para AMR gera uma perda considerável de informação, visto que quando ambas as sentenças de avaliação passam por tal processo, possuem uma semelhança maior do que quando apenas a sentença predita (ver Tabela 7).

No entanto, esta melhora confirmou-se parcialmente no modelo 5. Na avaliação com a medida METEOR, o modelo proposto obteve um resultado melhor do que o *baseline*. Porém, quando avaliado com a medida BLEU 1, o resultado do modelo foi inferior. Como a METEOR utiliza, além do alinhamento exato de palavras, o alinhamento de radical e de sinônimos, isso demonstra que as sentenças preditas pelo modelo 5 utilizam uma maior variação de palavras.

6.4.3.3 Resultados da Avaliação 3 – Sentenças de referência e desanonimizadas

A avaliação entre as sentenças de referência e desanonimizada teve como propósito avaliar as sentenças sem utilizar as sentenças preditas pelo modelo, ou seja, avaliar as sentenças de referência em LN (disponível no VG) e essas mesmas sentenças transformadas em AMR,

posteriormente anonimizadas e transformadas novamente em LN. Os resultados obtidos são exibidos na Tabela 9 e demonstram que apenas a desanonimização, resulta em uma perda maior que 50% na BLEU 1 e cerca de 75% quando avaliado com a METEOR.

Tabela 9 – Avaliação sentenças de referência e desanonimizadas

Modelo	BLEU 1	METEOR
1	0,486065	0,266540
2	0,439907	0,239266
3	0,478816	0,261192
4	0,487043	**0,267277**
5	**0,490973**	0,270997

Fonte: Elaborado pelo autor

Essa avaliação corrobora mais uma vez que apenas a transformação texto-AMR-texto, sem utilizar as sentenças preditas por algum dos modelos (1 a 5), já acarreta em uma perda significativa da qualidade das sentenças.

As medidas automáticas avaliam as sentenças do ponto da perspectiva sintática. No entanto, sentenças diferentes sintaticamente podem expressar a mesma ideia. Portanto, a fim de realizar uma análise comparativa também entre o conteúdo das sentenças, foi selecionado um pequeno conjunto de exemplos para a análise manual.

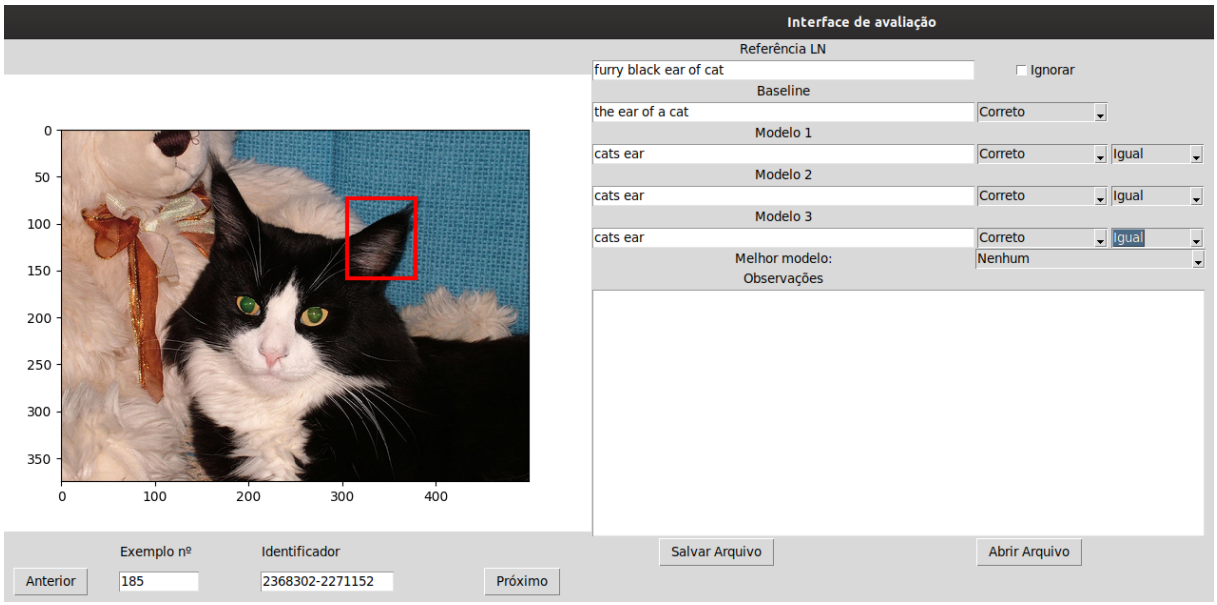
6.4.4 Análise manual

Apesar da avaliação automática ser extremamente útil e utilizada para comparação rápida e imparcial dos resultados, uma avaliação manual, realizada por humanos, pode oferecer uma visão diferente da qualidade dos modelos. Frases com palavras diferentes podem ser boas descrições para a região. Esse tipo de nuance entre as frases não pode ser captada pelas medidas automáticas.

Desse modo, neste trabalho também foi realizada uma análise manual de um conjunto de descrições geradas pelos modelos. Uma vez que uma avaliação manual de todas as sentenças preditas pelos 5 modelos, em todos os exemplos de teste, seria inviável, apenas uma pequena amostra de exemplos selecionados aleatoriamente foi analisada por uma pessoa não nativa da língua inglesa. Assim, as conclusões retiradas dessa análise manual devem ser interpretadas apenas como uma visão complementar dos resultados obtidos pelas medidas automáticas aplicadas a todo conjunto de teste.

A Figura 48 ilustra a interface desenvolvida para a avaliação manual. Nela, é possível carregar um arquivo com as descrições preditas pelo *baseline*(LN) e pelos modelos que usam AMR, nomeados na interface como Modelo 1 (AMR anonimizado), Modelo 2 (AMR anonimizado concatenado) e Modelo 3 (AMR anonimizado concatenado 2), além da referência (sentença original do VG).

Figura 48 – Interface de avaliação manual



Fonte: Elaborado pelo autor

Cada uma das descrições geradas pelos modelos treinados foi classificada em uma de três classes possíveis: correto, incorreto ou parcialmente correto. Descrições corretas são aquelas que trazem apenas informações corretas. Descrições parcialmente corretas são aquelas que trazem tanto informações corretas como erradas. Por fim, as descrições incorretas são aquelas que só trazem informações erradas. Vale mencionar que uma descrição foi avaliada em relação ao conteúdo que ela trazia e não à sua forma sintática. Assim, uma descrição que trouxe conteúdo correto em relação à *bbox*, mesmo que apresentasse erros sintáticos, foi marcada como correta.

Para os modelos em comparação, foi possível marcar se eles eram melhores, piores ou iguais ao *baseline*, e qual dos modelos gerou uma sentença melhor. Nessa comparação entre modelos, considerou-se a quantidade de informações corretas e relevantes apresentadas por cada um. Assim, se um modelo A trouxe mais informações corretas e relevantes do que um modelo B, então o modelo A foi considerado melhor. Se a quantidade de informações corretas e relevantes do modelo A foi menor, então o modelo B foi considerado melhor; e, em caso de empate, considerou-se ambos igualmente bons ou igualmente ruins.

A Figura 49 oferece um exemplo no qual o Modelo 2 foi considerado melhor porque trouxe uma informação correta e relevante a mais (“*white*”) do que os demais modelos em comparação. Além disso, todos os modelos em comparação foram considerados melhores do que o *baseline* porque trouxeram informação a mais do que ele (“*curtain*”).

Na interface é possível, ainda, ignorar o exemplo caso a referência em LN esteja incorreta o que, neste contexto, significa dizer que a referência não condiz com o que é apresentado na *bbox*. Um exemplo desse caso é apresentado na Figura 50.

Figura 49 – Exemplo de comparação entre modelos

Fonte: Elaborado pelo autor

Figura 50 – Exemplo de caso que foi ignorado na avaliação manual

Fonte: Elaborado pelo autor

Foram selecionados 400 exemplos aleatoriamente para a avaliação. Do total, 52 foram ignorados por não descreverem a *bbox* ou por terem diferenças pequenas em relação a outra *bbox* da mesma imagem também selecionada para avaliação manual.

A Tabela 10 apresenta os valores da análise manual. Os resultados demonstram que o número de total de exemplos corretos e parcialmente corretos do *baseline* (203 ou 58,32%) foram maiores que os modelos AMR anonimizado (182 ou 52,29%), AMR anonimizado concatenado

Tabela 10 – Análise manual de conjunto de amostras

	Avaliação das sentenças			Comparação com *baseline*		
Modelo	**Correto**	**Parc. Corr.**	**Incorreto**	**Melhor**	**Igual**	**Pior**
1	**145 (41,66%)**	58 (16,66%)	**145 (41,66%)**	-	-	-
3	104 (29,88%)	**78 (22,41%)**	166 (47,70%)	**83 (23,85%)**	148 (42,52%)	117 (33,62%)
4	122 (35,05%)	61 (17,52%)	165 (47,41%)	66 (18,96%)	**183 (52,58%)**	**99 (28,44%)**
5	109 (31,32%)	57 (16,37%)	182 (52,29%)	70 (20,11%)	166 (47,70%)	112 (32,18%)

Fonte: Elaborado pelo autor

(183 ou 52,56%) e AMR anonimizado concatenado 2 (166 ou 47,69%).

A partir dessa avaliação também é possível notar que para o modelo AMR anonimizado concatenado, 71,54% (249 exemplos) de suas sentenças foram melhores ou iguais ao *baseline*, contra 67,81% (236 exemplos) do modelo AMR anonimizado concatenado 2 e 66,37% (231 exemplos) para o modelo AMR anonimizado.

As sentenças do modelo AMR anonimizado foram consideradas as melhores sentenças preditas pelos modelos AMR em 16,95% (59) dos casos, contra 11,78% (41) das sentenças do modelos AMR anonimizado concatenado e 10,91% (38) do modelo AMR anonimizado concatenado 2.

Essa análise manual mostra que os resultados obtidos pelas medidas automáticas podem não refletir a qualidade das sentenças preditas pelos modelos quando avaliadas por humanos. Outro exemplo é ilustrado na Figura 51 onde apesar da sentença gerada por cada um dos modelos serem diferentes, referindo-se cada uma a elementos diferentes da *bbox*, todas foram consideradas corretas, e igualmente boas em relação a sentença de referência.

Figura 51 – Exemplo de sentenças preditas diferentes da referência e corretas

Fonte: Elaborado pelo autor

Capítulo 7

Conclusões e Trabalhos Futuros

Este trabalho abordou a geração de descrições para regiões de imagem, sendo esta uma tarefa emergente na literatura em que estão sendo propostas novas estratégias em busca de modelos computacionais que produzam sentenças próximas a de humanos.

Este trabalho motivou-se a partir de Karpathy et al. (2014), no qual foram utilizadas relações de dependências para a tarefa de recuperação informação. Até então, os trabalhos da literatura utilizavam a língua natural em sua forma pura, ou seja, sem enriquecimento de qualquer espécie. A utilização das relações de dependências demonstrou ser benéfica na recuperação, com resultados superiores aos trabalhos até então.

A partir da conclusão de Karpathy et al. (2014), este trabalho motivou-se a utilizar recursos de outro nível de informação, o nível semântico. Para isso, foi escolhido utilizar a representação semântica AMR (BANARESCU et al., 2013), que tem como finalidade unificar a maioria das tarefas semânticas, representando de uma única forma a maioria dos fenômenos linguísticos, procurando representar sentenças com mesmo sentido de uma única forma, mesmo que diferentes sintaticamente.

Assim como na recuperação da informação, trabalhos de GDRI também utilizam a língua natural em sua forma pura. Os trabalhos de GDRI fazem uso extenso das arquiteturas propostas para a detecção de objetos, principalmente a *Faster RCNN* (Ren et al., 2017), adaptando-as para que possam ser capazes de, agora não detectar objetos, mas produzir uma sentença que descreva a região.

A arquitetura da *Faster RCNN*, para a detecção de objetos, é composta por uma CNN, que possui como entrada a imagem completa para extração das características. Então, a RPN propõe regiões as quais a RoI opera para seleção. Por fim, uma camada totalmente conectada processa as RoI, para posteriormente ocorrer a classificação, através de uma função *softmax*, e a regressão da *bbox*.

Nos últimos, alguns trabalhos foram propostos que significaram avanços para a área Karpathy e Fei-Fei (2015), Zhang et al. (2015), Johnson et al. (2016), Mao et al. (2016), Yu et al. (2016), Yang et al. (2017), Yin et al. (2019) e Zhang et al. (2019). O modelo proposto

por Johnson et al. (2016) utiliza uma técnica de interpolação bilinear para o ajuste da *bbox*. Por esse ter mostrado-se promissor, e seu código estar disponível, foi escolhido este modelo para a experimentação.

Para o trabalho de Johnson et al. (2016), realizou-se algumas alterações. Após a CNN, existe uma camada de localização em que, de maneira geral, utiliza interpolação bilinear para propor melhores regiões. E por fim, a LSTM produz uma sentença a partir de cada uma das regiões.

A partir do trabalho de Johnson et al. (2016), foram realizados treinamentos com 3 formas de representação da sentença baseados na AMR: AMR linearizada; AMR anonimizada; AMR anonimizada concatenada. Mais detalhes podem ser vistos na Seção 5.2. Um outra versão da AMR anonimizada concatenada também foi treinada, mas com os parâmetros diferentes, para manter uma proporção de imagens, sentenças e *tokens* mais próximos ao *baseline* (Seção 6.3.1).

A avaliação foi dividida de duas formas: de maneira automática no conjunto completo de teste (cerca de 5.000 exemplos) e manual (em cerca de 350 exemplos do conjunto de teste, selecionados aleatoriamente). A primeira avaliação automática ocorreu com base no cálculo da mAP (principal medida de avaliação em GDRI) sobre exatamente as sentenças que foram preditas pelos modelos, ou seja, sobre a sua respectiva forma de representação. Com exceção do modelo AMR linearizado (que mostrou-se competitivo), todos os demais foram consideravelmente superiores ao *baseline* (Seção 6.4.1).

Para as outras avaliações automáticas, para cada um dos modelos, ocorreram em LN e também em AMR (através da medida SMatch), exceto este último para o *baseline*, da seguinte forma: sentenças de referência e preditas; sentenças desanonimizadas e preditas; e sentenças de referência e desanonimizadas. Na avaliação com a SMATCH (Tabela 6), o modelo AMR anonimizado obteve o melhor resultado na avaliação entre a referência e predita. Para as sentenças desanonimizadas e preditas, o modelo AMR anonimizado concatenado se saiu melhor, enquanto para as as referências e desanonimizadas o melhor resultado ocorreu para o modelo AMR. Nessas avaliações vale destacar o impacto do ruído que as ferramentas de conversão texto-AMR e AMR-texto trazem aos resultados, em média de 21%, como pode ser observado na coluna referência e desanonimizada.

Para a avaliação em LN, realizada com a BLEU 1 e METEOR, as sentenças preditas pelos modelos em AMR foram devidamente transformados em LN. Na avaliação entre as sentenças de referência e preditas (Tabela 7) o *baseline* teve melhor resultado. No entanto, quando avaliado entre as sentenças desanonimizadas e preditas (Tabela 8), e de referência e desanonimizadas (Tabela 9), o modelo AMR anonimizado concatenado resultou em um valor maior. Nessas avaliações vale destacar o impacto do ruído que as ferramentas de conversão texto-AMR e AMR-texto trazem aos resultados, de mais de 50% na BLEU 1 e de cerca de 75% na METEOR, observado na avaliação referência e desanonimizada.

Isso corrobora que, mesmo com um resultado inferior na avaliação entre as sentenças de referência e preditas, a AMR, em especial na sua versão anonimizada concatenada, pode beneficiar as descrições de regiões de imagens, visto que nas outras duas avaliações automáticas, os resultados foram superiores.

Na análise manual, realizada por uma pessoa não nativa da língua inglesa (Tabela 10), o *baseline* ainda mostrou-se superior. Do total de exemplos avaliados, 58,32% das sentenças foram classificadas como corretas ou parcialmente corretas para o *baseline*, em comparação com 52,29% do modelo AMR anonimizado, 52,56% do modelo AMR anonimizado concatenado e 47,69% do modelo AMR anonimizado concatenado 2. Em relação aos três modelos avaliados, as sentenças do modelo AMR anonimizado concatenado foram consideradas melhores ou iguais ao *baseline* em 71,54%, 67,81% do modelo AMR anonimizado concatenado 2 e 66,37% do modelo anonimizado.

Por fim, vale ressaltar que a única medida de avaliação que leva em consideração a imagem juntamente com a descrição gerada para ela é a mAP. Todas as outras medidas de avaliação usadas (SMATCH, BLEU-1 e METEOR) consideram apenas a semelhança entre a sentença predita e a sentença de referência e, como ilustrado no exemplo da Figura 51, em GDRI nem sempre a sentença de referência é a única possível para descrever uma dada região da imagem. Assim, como já mencionado, a medida automática que melhor retrata o desempenho dos modelos avaliados é a mAP e, segundo esta medida, a AMR mostrou-se efetiva para a GDRI. Contudo, há a necessidade de melhorar a qualidade das ferramentas de conversão texto-AMR e AMR-texto uma vez que o ruído que elas agregam aos resultados é bastante impactante.

7.1 Trabalhos futuros

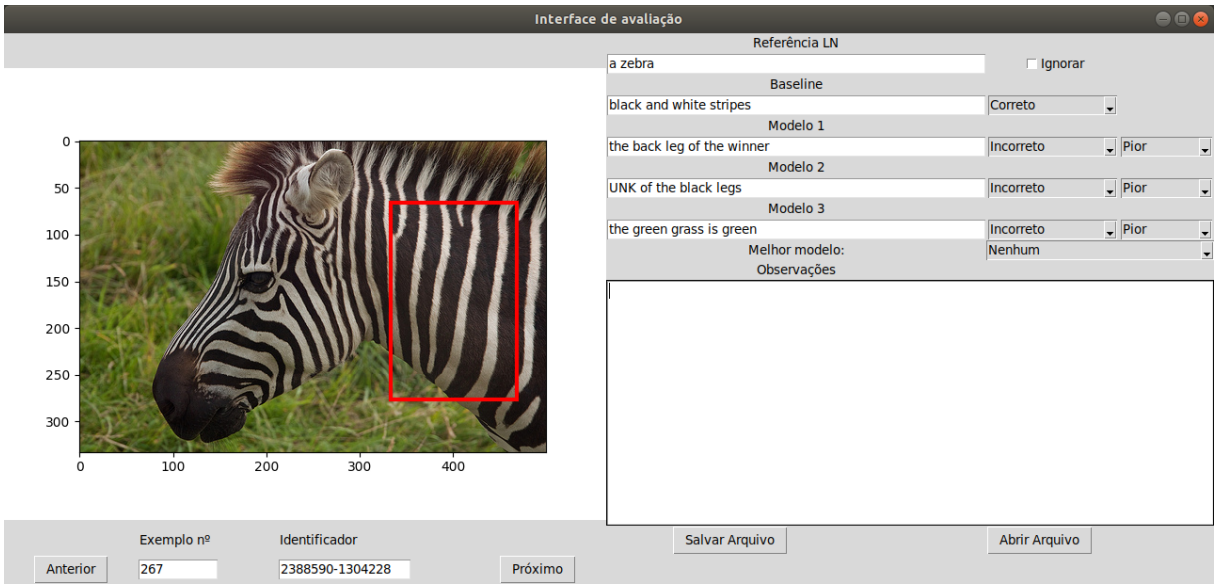
Nessa seção, serão detalhados trabalhos futuros que podem ser realizados a partir deste projeto.

7.1.1 Treinamento dos *parsers* AMR

Os *parsers* utilizados neste trabalho, que convertem a sentença de LN para AMR (*Text-to-AMR*) e de AMR para LN (AMR-to-Text), que norteiam a utilização da RS AMR foram treinados em outros conjuntos de dados. Isso prejudicou a conversão das sentenças entre as representações, pois palavras desconhecidas fora do escopo do treinamento podem ter sido utilizadas na execução dos mesmos.

Um erro recorrente encontrado na transformação das sentenças em AMR e LN é a ocorrência sistemática de palavras que não estão presentes na sentença em AMR e que não podem ser inferidas através da semântica. Um exemplo é ilustrado na Figura 52, em que houve a ocorrência da palavra *winner*.

Figura 52 – Exemplo de imagem do conjunto de teste



Fonte: Elaborado pelo autor

A partir dessa problemática, um trabalho futuro para este projeto seria treinar os *parsers* no mesmo conjunto de dados utilizado para a geração das descrições para as regiões de imagem. Isso pode resultar em uma melhora significativa da qualidade das sentenças convertidas de AMR para língua natural.

7.1.2 Experimentação em outros modelos

Outro trabalho futuro é a experimentação das formas de representações das sentenças usando AMR (Seção 5.2) em outros modelos de GDRI. Um dos objetivos iniciais desse trabalho era realizar os mesmos experimentos que foram feitos com o modelo de Johnson et al. (2016), no modelo de Yang et al. (2017). Contudo, uma vez que o treinamento com o modelo neural de Johnson et al. (2016) tomou mais tempo do que o previsto, essa experimentação não foi possível. Os trabalhos de Zhang et al. (2019) e Yin et al. (2019), que obtiveram resultados superiores a Johnson et al. (2016), foram publicados no final do desenvolvimento deste projeto, também não permitindo tempo suficiente para sua utilização, uma vez que o seus códigos fonte também ainda não foram disponibilizados.

Dessa forma, indica-se esses trabalhos como experimentação futuras por dois motivos. O primeiro é prover uma maior experimentação em relação ao número de modelos verificados na capacidade dos mesmo em produzir sentenças em algum formato AMR ou derivado. O segundo motivo é que tais modelo mostraram-se superiores ao de Johnson et al. (2016), sendo extremamente promissores para a GRDI com AMR.

7.1.3 Criação de medida específica para avaliação

Como explicado na Seção 3.4, as medidas utilizadas para avaliação dos modelos de GDRI não foram criadas com este intuito, sendo adaptadas para esta tarefa. A BLEU e a METEOR foram criadas para avaliarem a qualidade de modelos de tradução automática, e a mAP originou-se na detecção de objetos.

De modo geral, tais medidas necessitam que as mesmas palavras, radicais ou sinônimos sejam usados para que valores maiores sejam obtidos. No entanto, conforme demonstrado na análise manual (Seção 6.4.4), sentenças com palavras diferentes e descrevendo diferentes objetos da região, também podem ser corretas. Desse modo, um trabalho futuro, que também beneficiaria toda a área, seria o desenvolvimento de uma medida de avaliação automática específica para a GDRI, considerando a maior parte dos fatores utilizados pelos seres humanos.

7.2 Contribuições

As principais contribuições deste trabalho podem ser resumidas em:

1. Utilização da principal representação semântica da atualidade, AMR, para a tarefa de GDRI.
2. Construção do primeiro córpus AMR para GDRI, e possivelmente, também o primeiro córpus imagem-AMR.
3. Comprovação da hipótese de que a utilização de sentenças em AMR e suas variações pode beneficiar a GDRI em comparação com o uso de LN pura.
4. Constatação da limitação das atuais medidas de avaliação, adaptadas de outras tarefas, aplicadas na GDRI, uma vez que elas não retratam toda a qualidade da descrição predita em relação a sua *bbox*.
5. Disponibilização pública dos *scripts* e córpus usados no treinamento dos modelos no repositório do LALIC[1].

[1] https://github.com/LALIC-UFSCar/densecap-amr

REFERÊNCIAS

ABEND, O.; RAPPOPORT, A. The state of the art in semantic representation. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2017. v. 1, p. 77–89. Citado na página 24.

ALUÍSIO, S. M.; SPECIA, L.; PARDO, T. A.; MAZIERO, E. G.; FORTES, R. P. Towards brazilian portuguese automatic text simplification systems. In: *Proceedings of the Eighth ACM Symposium on Document Engineering*. New York, NY, USA: ACM, 2008. (DocEng '08), p. 240–248. ISBN 978-1-60558-081-4. Disponível em: <http://doi.acm.org/10.1145/1410140.1410191>. Citado na página 29.

ANCHIÊTA, R.; PARDO, T. Towards AMR-BR: A sembank for brazilian portuguese language. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association, 2018. Disponível em: <http://aclweb.org/anthology/L18-1157>. Citado 5 vezes nas páginas 15, 16, 24, 25 e 26.

BANARESCU, L.; BONIAL, C.; CAI, S.; GEORGESCU, M.; GRIFFITT, K.; HERMJAKOB, U.; KNIGHT, K.; KOEHN, P.; PALMER, M.; SCHNEIDER, N. Abstract meaning representation for sembanking. In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. [S.l.: s.n.], 2013. p. 178–186. Citado 4 vezes nas páginas 15, 24, 33 e 90.

BANERJEE, S.; LAVIE, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, 2005. p. 65–72. Disponível em: <https://www.aclweb.org/anthology/W05-0909>. Citado na página 36.

BOJAR, O.; BUCK, C.; CALLISON-BURCH, C.; HADDOW, B.; KOEHN, P.; MONZ, C.; POST, M.; SAINT-AMAND, H.; SORICUT, R.; SPECIA, L. (Ed.). *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, 2013. Disponível em: <https://www.aclweb.org/anthology/W13-2200>. Citado na página 34.

BROWN, J. C.; FRISHKOFF, G. A.; ESKENAZI, M. Automatic question generation for vocabulary assessment. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005. (HLT '05), p. 819–826. Disponível em: <https://doi.org/10.3115/1220575.1220678>. Citado na página 29.

CAI, S.; KNIGHT, K. Smatch: an evaluation metric for semantic feature structures. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 748–752.

Disponível em: <https://www.aclweb.org/anthology/P13-2131>. Citado 2 vezes nas páginas 41 e 43.

CASELI, H. M.; NUNES, M. D.; FORCADA, M. L. Automatic induction of bilingual resources from aligned parallel corpora: Application to shallow-transfer machine translation. *Machine Translation*, Kluwer Academic Publishers, Hingham, MA, USA, v. 20, n. 4, p. 227–245, dez. 2006. ISSN 0922-6567. Disponível em: <http://dx.doi.org/10.1007/s10590-007-9027-9>. Citado na página 29.

CHOWDHURY, G. G. Natural language processing. *Annual Review of Information Science and Technology (ARIST)*, v. 37, p. 51–89, 2003. ISSN 0066-4200. Citado na página 28.

CORTES, C.; VAPNIK, V. Support-vector networks. In: *Machine Learning*. [S.l.: s.n.], 1995. p. 273–297. Citado na página 19.

DAMONTE, M.; COHEN, S. B. Cross-lingual abstract meaning representation parsing. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. p. 1146–1155. Disponível em: <http://aclweb.org/anthology/N18-1104>. Citado na página 25.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919. Citado 2 vezes nas páginas 45 e 49.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J. M.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, v. 88, p. 303–338, 2009. Citado na página 38.

GARCÍA-MÉNDEZ, S.; FERNÁNDEZ-GAVILANES, M.; COSTA-MONTENEGRO, E.; JUNCAL-MARTÍNEZ, J.; GONZÁLEZ-CASTAÑO, F. J. A library for automatic natural language generation of spanish texts. *Expert Systems with Applications*, v. 120, p. 372 – 386, 2019. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417418307565>. Citado 2 vezes nas páginas 13 e 28.

GATT, A.; KRAHMER, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 61, n. 1, p. 65–170, jan. 2018. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=3241691.3241693>. Citado na página 29.

GIRSHICK, R. Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1440–1448. ISSN 2380-7504. Citado na página 19.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2014. (CVPR '14), p. 580–587. ISBN 978-1-4799-5118-5. Disponível em: <https://doi.org/10.1109/CVPR.2014.81>. Citado 2 vezes nas páginas 18 e 49.

He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2017. p. 2980–2988. Citado na página 62.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997. Disponível em: <https://doi.org/10.1162/neco.1997.9.8.1735>. Citado na página 23.

HODOSH, M.; YOUNG, P.; HOCKENMAIER, J. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 47, n. 1, p. 853–899, maio 2013. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=2566972.2566993>. Citado na página 45.

JOHNSON, J.; KARPATHY, A.; FEI-FEI, L. Densecap: Fully convolutional localization networks for dense captioning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. Citado 25 vezes nas páginas 11, 12, 40, 44, 50, 51, 57, 59, 60, 61, 62, 64, 65, 66, 67, 68, 74, 76, 77, 78, 79, 80, 90, 91 e 93.

JURAFSKY, D.; MARTIN, J. H. *Speech and language processing*. [S.l.: s.n.], 2018. v. 3. Citado 2 vezes nas páginas 13 e 24.

KARPATHY, A. *Andrej Karpathy blog*. GitHub, 2015. Disponível em: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>. Citado 2 vezes nas páginas 21 e 22.

KARPATHY, A.; FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. Citado 8 vezes nas páginas 11, 23, 44, 46, 47, 48, 65 e 90.

KARPATHY, A.; JOULIN, A.; FEI-FEI, L. Deep fragment embeddings for bidirectional image sentence mapping. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 1889–1897. Disponível em: <http://dl.acm.org/citation.cfm?id=2969033.2969038>. Citado 8 vezes nas páginas 11, 15, 44, 45, 46, 57, 65 e 90.

KAUCHAK, D.; BARZILAY, R. Paraphrasing for automatic evaluation. In: *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. (HLT-NAACL '06), p. 455–462. Disponível em: <https://doi.org/10.3115/1220835.1220893>. Citado na página 29.

KAZEMZADEH, S.; ORDONEZ, V.; MATTEN, M.; BERG, T. L. ReferIt game: Referring to objects in photographs of natural scenes. In: *EMNLP*. [S.l.: s.n.], 2014. Citado na página 56.

KONSTAS, I.; IYER, S.; YATSKAR, M.; CHOI, Y.; ZETTLEMOYER, L. Neural AMR: Sequence-to-sequence models for parsing and generation. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 146–157. Disponível em: <https://www.aclweb.org/anthology/P17-1014>. Citado na página 69.

KRISHNA, R.; ZHU, Y.; GROTH, O.; JOHNSON, J.; HATA, K.; KRAVITZ, J.; CHEN, S.; KALANTIDIS, Y.; LI, L.-J.; SHAMMA, D. A.; BERNSTEIN, M. S.; FEI-FEI, L. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, Kluwer Academic Publishers, USA, v. 123, n. 1, p. 32–73, maio 2017. ISSN 0920-5691. Disponível em: <https://doi.org/10.1007/s11263-016-0981-7>. Citado 6 vezes nas páginas 14, 48, 49, 73, 75 e 76.

LIAO, K.; LEBANOFF, L.; LIU, F. Abstract Meaning Representation for multi-document summarization. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 1178–1190. Disponível em: <https://www.aclweb.org/anthology/C18-1101>. Citado 4 vezes nas páginas 8, 15, 32 e 33.

LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004. p. 74–81. Disponível em: <https://www.aclweb.org/anthology/W04-1013>. Citado na página 33.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014. p. 740–755. ISBN 978-3-319-10602-1. Citado na página 74.

LIU, F.; FLANIGAN, J.; THOMSON, S.; SADEH, N.; SMITH, N. A. Toward abstractive summarization using semantic representations. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, 2015. p. 1077–1086. Disponível em: <https://www.aclweb.org/anthology/N15-1114>. Citado na página 32.

LYU, C.; TITOV, I. AMR parsing as graph prediction with latent alignment. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 32 e 68.

MACDONALD, I.; SIDDHARTHAN, A. Summarising news stories for children. In: *Proceedings of the 9th International Natural Language Generation conference*. [S.l.: s.n.], 2016. p. 1–10. Citado na página 29.

MAO, J.; HUANG, J.; TOSHEV, A.; CAMBURU, O.; YUILLE, A.; MURPHY, K. Generation and comprehension of unambiguous object descriptions. In: *CVPR*. [S.l.: s.n.], 2016. Citado 10 vezes nas páginas 11, 44, 52, 53, 54, 55, 56, 57, 65 e 90.

MATTHIESSEN, C.; BATEMAN, J. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter, 1991. (Communication in artificial intelligence). ISBN 9780861877119. Disponível em: <https://books.google.com.br/books?id=f_RhAAAAMAAJ>. Citado na página 25.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, v. 2013, 01 2013. Citado 2 vezes nas páginas 47 e 74.

PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W.-J. BLEU: A method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ACL '02), p. 311–318. Disponível em: <https://doi.org/10.3115/1073083.1073135>. Citado 2 vezes nas páginas 35 e 74.

PARDO, T. A. S.; RINO, L. H. M.; NUNES, M. d. G. V. Gistsumm: A summarization tool based on a new extractive method. In: MAMEDE, N. J.; TRANCOSO, I.; BAPTISTA, J.; NUNES, M. das G. V. (Ed.). *Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 210–218. ISBN 978-3-540-45011-5. Citado na página 29.

RASHTCHIAN, C.; YOUNG, P.; HODOSH, M.; HOCKENMAIER, J. Collecting image annotations using amazon's mechanical turk. In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Los Angeles: Association for Computational Linguistics, 2010. p. 139–147. Disponível em: <https://www.aclweb.org/anthology/W10-0721>. Citado na página 45.

REITER, E.; DALE, R. Building applied natural language generation systems. *Natural Language Engineering*, Cambridge University Press, v. 3, n. 1, p. 57–87, 1997. Citado 2 vezes nas páginas 13 e 28.

REITER, E.; DALE, R. *Building Natural Language Generation Systems*. New York, NY, USA: Cambridge University Press, 2000. ISBN 0-521-62036-8. Citado na página 28.

REITER, E.; SRIPADA, S.; HUNTER, J.; YU, J.; DAVY, I. Choosing words in computer-generated weather forecasts. *Artif. Intell.*, Elsevier Science Publishers Ltd., Essex, UK, v. 167, n. 1-2, p. 137–169, set. 2005. ISSN 0004-3702. Disponível em: <https://doi.org/10.1016/j.artint.2005.06.006>. Citado na página 29.

Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 6, p. 1137–1149, June 2017. ISSN 0162-8828. Citado 7 vezes nas páginas 20, 21, 52, 59, 61, 62 e 90.

ROSEBROCK, A. *Intersection over Union (IoU) for object detection*. [S.l.]: Google Patents, 2020. [Online]. Disponível em: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection>. Citado na página 39.

RUS, V.; WYSE, B.; PIWEK, P.; LINTEAN, M.; STOYANCHEV, S.; MOLDOVAN, C. The first question generation shared task evaluation challenge. In: *Proceedings of the 6th International Natural Language Generation Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (INLG '10), p. 251–257. Disponível em: <http://dl.acm.org/citation.cfm?id=1873738.1873777>. Citado na página 29.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015. Citado na página 19.

SIDDHARTHAN, A. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, John Benjamins Publishing Company, v. 165, n. 2, p. 259–298, 2014. Citado na página 29.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2015. Citado 3 vezes nas páginas 49, 51 e 62.

SONG, L.; ZHANG, Y.; WANG, Z.; GILDEA, D. A graph-to-sequence model for AMR-to-text generation. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL-18)*. Melbourne, Australia: [s.n.], 2018. Citado na página 32.

SPECIA, L.; RINO, L. H. M. *Representação Semântica: Alguns Modelos Ilustrativos*. [S.l.], 2002. Disponível em: <http://wiki.icmc.usp.br/images/1/1c/SpeciaRino2002.pdf>. Citado 2 vezes nas páginas 14 e 24.

SRIVASTAVA, G.; SRIVASTAVA, R. A survey on automatic image captioning. In: GHOSH, D.; GIRI, D.; MOHAPATRA, R. N.; SAVAS, E.; SAKURAI, K.; SINGH, L. P. (Ed.). *Mathematics and Computing*. Singapore: Springer Singapore, 2018. p. 74–83. ISBN 978-981-13-0023-3. Citado 2 vezes nas páginas 29 e 30.

STEINBACH, M.; KARYPIS, G.; KUMAR, V. A comparison of document clustering techniques. *Proceedings of the International KDD Workshop on Text Mining*, 06 2000. Citado na página 74.

SUMATHI, T.; HEMALATHA, M. A combined hierarchical model for automatic image annotation and retrieval. In: *2011 Third International Conference on Advanced Computing*. [S.l.: s.n.], 2011. p. 135–139. ISSN 2377-6927. Citado na página 29.

TAMCHYNA, A.; QUIRK, C.; GALLEY, M. A discriminative model for semantics-to-string translation. In: *Proceedings of the 1st Workshop on Semantics-Driven Statistical Machine Translation (S2MT 2015)*. Beijing, China: Association for Computational Linguistics, 2015. p. 30–36. Disponível em: <https://www.aclweb.org/anthology/W15-3504>. Citado 4 vezes nas páginas 15, 32, 33 e 34.

THOMEE, B.; SHAMMA, D. A.; FRIEDLAND, G.; ELIZALDE, B.; NI, K.; POLAND, D.; BORTH, D.; LI, L.-J. Yfcc100m: The new data in multimedia research. *Commun. ACM*, ACM, New York, NY, USA, v. 59, n. 2, p. 64–73, jan. 2016. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2812802>. Citado 2 vezes nas páginas 52 e 74.

UIJLINGS, J. R. R.; SANDE, K. E. A. van de; GEVERS, T.; SMEULDERS, A. W. M. Selective search for object recognition. *International Journal of Computer Vision*, v. 104, n. 2, p. 154–171, Sep 2013. ISSN 1573-1405. Disponível em: <https://doi.org/10.1007/s11263-013-0620-5>. Citado na página 19.

VANDERWENDE, L.; MENEZES, A.; QUIRK, C. An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Denver, Colorado: Association for Computational Linguistics, 2015. p. 26–30. Disponível em: <https://www.aclweb.org/anthology/N15-3006>. Citado 2 vezes nas páginas 33 e 34.

VIETHEN, J.; DALE, R. Algorithms for generating referring expressions: Do they do what people do? In: *Proceedings of the Fourth International Natural Language Generation Conference*. Association for Computational Linguistics, 2006. p. 63–70. Disponível em: <http://aclweb.org/anthology/W06-1410>. Citado na página 52.

WANG, C.; LI, B.; XUE, N. Transition-based chinese amr parsing. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 2018. p. 247–252. Disponível em: <http://aclweb.org/anthology/N18-2040>. Citado na página 24.

WANG, C.; YANG, H.; BARTZ, C.; MEINEL, C. Image captioning with deep bidirectional lstms. In: *Proceedings of the 24th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2016. (MM '16), p. 988–997. ISBN 978-1-4503-3603-1. Disponível em: <http://doi.acm.org/10.1145/2964284.2964299>. Citado na página 30.

WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q. V.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K.; KLINGNER, J.; SHAH, A.; JOHNSON, M.; LIU, X.;

KAISER Łukasz; GOUWS, S.; KATO, Y.; KUDO, T.; KAZAWA, H.; STEVENS, K.; KURIAN, G.; PATIL, N.; WANG, W.; YOUNG, C.; SMITH, J.; RIESA, J.; RUDNICK, A.; VINYALS, O.; CORRADO, G.; HUGHES, M.; DEAN, J. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. Citado na página 29.

YAGCIOGLU, S.; ERDEM, E.; ERDEM, A.; CAKICI, R. A distributed representation based query expansion approach for image captioning. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, 2015. p. 106–111. Disponível em: <http://aclweb.org/anthology/P15-2018>. Citado na página 29.

YANG, L.; TANG, K.; YANG, J.; LI, L.-J. Dense captioning with joint inference and visual context. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. Citado 13 vezes nas páginas 11, 31, 44, 57, 58, 59, 60, 61, 62, 64, 65, 90 e 93.

YIN, G.; SHENG, L.; LIU, B.; YU, N.; WANG, X.; SHAO, J. Context and attribute grounded dense captioning. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. Citado 9 vezes nas páginas 9, 11, 60, 61, 62, 64, 65, 90 e 93.

YOUNG, P.; LAI, A.; HODOSH, M.; HOCKENMAIER, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, v. 2, p. 67–78, 2014. ISSN 2307-387X. Disponível em: <https://transacl.org/ojs/index.php/tacl/article/view/229>. Citado na página 45.

YU, L.; POIRSON, P.; YANG, S.; BERG, A. C.; BERG, T. L. Modeling context in referring expressions. In: LEIBE, B.; MATAS, J.; SEBE, N.; WELLING, M. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 69–85. ISBN 978-3-319-46475-6. Citado 6 vezes nas páginas 11, 44, 55, 56, 65 e 90.

ZHANG, X.; SONG, X.; LV, X.; JIANG, S.; YE, Q.; JIAO, J. Rich image description based on regions. In: *Proceedings of the 23rd ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2015. (MM '15), p. 1315–1318. ISBN 9781450334594. Disponível em: <https://doi.org/10.1145/2733373.2806338>. Citado 8 vezes nas páginas 8, 11, 13, 31, 44, 49, 65 e 90.

ZHANG, Z.; ZHANG, Y.; SHI, Y.; YU, W.; NIE, L.; HE, G.; FAN, Y.; YANG, Z. Dense image captioning based on precise feature extraction. In: GEDEON, T.; WONG, K. W.; LEE, M. (Ed.). *Neural Information Processing*. Cham: Springer International Publishing, 2019. p. 83–90. ISBN 978-3-030-36802-9. Citado 9 vezes nas páginas 9, 11, 60, 62, 63, 64, 65, 90 e 93.