University of Amsterdam

PhD Thesis

Multi-modal Learning Algorithms for Sequence Modeling and Representation Learning

Maurits Bleeker

Multi-modal Learning Algorithms for Sequence Modeling and Representation Learning

Maurits Bleeker

Multi-modal Learning Algorithms for Sequence Modeling and Representation Learning

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op vrijdag 14 juni 2024, te 14:00 uur

door

Maurits Johannes Robertson Bleeker

geboren te Hoorn

PROMOTIECOMMISSIE

Promotor:

prof. dr. M. de Rijke	Universiteit van Amsterdam

Copromotor:

dr. A.C. Yates	Universiteit van Amsterdam

Overige leden:

prof. dr. E. Kanoulas	Universiteit van Amsterdam
prof. dr. M.F. Moens	KU Leuven
dr. S. Pezzelle	Universiteit van Amsterdam
prof. dr. C.G.M. Snoek	Universiteit van Amsterdam
prof. dr. M. Worring	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The work described in this thesis has been primarily carried out at the Information Retrieval Lab of the University of Amsterdam and in part during an internship at Apple AI/ML. The research carried out at the University of Amsterdam was funded by the Nederlandse Politie.

Printed by: Gildeprint – The Netherlands.

ISBN: 978-94-6496-130-0

Acknowledgements

On November 1, 2018, I started my PhD. I knew that it would be an exciting journey. However, what I did not fully realize at the time, was that it would be the people around me that would make the experience truly amazing and enjoyable from start to finish. I am grateful to everyone who has been part of this journey.

First and foremost, *Maarten* – I cannot express how thankful I am for the opportunity you gave me to pursue a PhD under your supervision. You have helped me to grow as a person. Thank you for giving me the freedom to explore every *modality* of a PhD, by helping me to build up my career, becoming a researcher, and a teacher.

Andrew – thank you for being my co-supervisor and for always making time to discuss research with me. Whenever a very (vague) new idea popped into my head, I could not wait to discuss it with you and hear your opinion about it. I have learned a lot from you over the last two years.

I am very honored to have *Cees*, *Evangelos*, *Marcel*, *Marie-Francine*, and *Sandro* serving on my PhD committee. Thank you for critically evaluating my thesis.

The IRLab – I do not think there are many academic research labs where you could have a better start to your research career than in IRLab. I would like to thank everyone who has been part of the lab for creating this fantastic environment and making Science Park such a nice place to work. Thank you for all the fun research meetings, Soos talks, endless conversations about both research and non-research, after-work drinks, coffee breaks, lunch at Cafe 6 and the main building, and the countless bitterballen and fries at Oerknal and elsewhere.

Philipp – I never thought you would say yes to the ridiculous idea of working out every morning before work in Westerpark, but you did, and I have met a new friend during those countless workouts (and the "burning bridges," of course). Thank you.

I would also like to thank my non-IRlab friends: *David* & *Sarah* – thank you for being there and for all the discussions about academia, internships, and life in general.

I want to thank the students I worked with, who have taught me sometimes more than I taught them.

I want to thank everyone I worked with at the *Dutch Police*. *Jon*, *Martin*, and *Peter* – it was unknown territory we all went into, but we always managed to make it work. Thank you for your patience and support. *Arthur*, *Dominique*, *Emil*, *Hilde*, *Jorn*, and *Nils* – thank you for being part of the Police Lab AI journey.

Gerard – thank you for helping me at the very beginning of this journey. Without you teaching me the math foundations that I needed for my master's, this whole thesis would have never been written.

Jelle & *Tim* – thank you for giving me my first job in the tech world. You are among the best managers a student and beginner programmer can wish for. Thanks for creating such an amazing environment and for giving me the opportunity to work with you.

Ana & *Patrick* – thank you for agreeing to be my paranymphs. *Ana* – thank you for being there for me since the beginning of my PhD journey. We collaborated, wrote multiple papers, supervised students, and even created an entire course together! Who better could I have asked to help with the very last step of this PhD? Thank you for all the great chats, support, and fun we have! *Patrick* – if there is anyone I trust to get anything done, it is you. Who better to stand with me as I defend my thesis? Thank you for everything, but mainly for being such a good friend. Oh, and Pat, π.

Maartje – I know no one in this world who is better at the *travelling salesman problem* than you. However, instead of always finding the *shortest* possible route, I think you always know how to find the *best* possible routes and share them with the people around you. Thank you for being part of this journey already since the master's, and for the great (and extensive) conversations we have.

De Hazekoppe; *Beu*, *Broer*, *Doni*, *Ess*, *FJ*, *Jor*, *Kaasie*, *Koen*, *Marrak*, *Pat*, *Pé*, *Roy*, *Skedel*, *Struul*, *Ti*, and *Yous* – I do not even know where to start. How do you summarise more than 14 years of friendship? Maybe we played a bit too much stress-pong (and too loud ...), poker, RISK, or rudo. Or maybe, we broke a rule or two during the COVID lockdowns. Thank you all for all the fun we have, the countless holidays/weekend trips, festivals, and parties. For not asking any questions about what I was actually doing at work and for keeping me away from the "kompjotr" and letting me "lekker rommelen". But most importantly, for all the support. Our friendship means a lot to me.

And next, of course, I would also like to thank my *spin-off Hazekoppe friends*; *Bren*, *Britt*, *Eer*, *Gu* (#LL zinin), *Liza*, *Roy*, and *Tim* (hoe staat de voorbereiding er voor?) – thank you for all the fun!

IR2 + BJT; *Bas*, *Joost*, *Jörg*, *Maartje*, *Thijs*, and *Ties* – thank you for all the dinners and drinks, being my AI friends, and discussing life and everything beyond. *Ties* – thank you for calling me at the most random moments of the day to start the best conversations. *Jörg* – you showed me how beautiful it is to stay learning your entire life. Thank you for being such an inspiration and always listening to the things I have to say.

In the summer of 2022, I spent four months at Apple AI/ML. *Pawel* – thank you for being such a fantastic mentor, writing a paper together, and introducing me to the ASR field. *Xiaodan* – thank you for welcoming me to your team. *Thijs* – thank you for letting me stay with you for a month and for the off-work fun.

During my three and half months at Amazon in 2023, I had the pleasure of working with the fantastic language modeling team. *Volker* – thank you for building such an amazing team and being a great manager. *Maciej* – for being my mentor/manager, and for all the good discussions we had. *Arne* – for letting me be the second hardest working intern, and thanks to *Aman*, *Leif*, and *Jens* for all the support and discussions about the project. *Nitish* – thank you for being "the real OG" favorite roommate!

Finally, I would like to thank my family. *Ome Koos* – thank you for moving me in and out of every apartment I have lived in so far in my life. *Arnoud* & *Koen* – thank you for being my *epic* brothers. Thanks for all the countless "potjes Catan en Keezen", and for always beating me with more than a minute on both the Anntal (i.e., La Longia) and Saslong slopes. But most importantly, for all the much-needed distraction from work and fun. *Mam* – your proactiveness has always inspired me! Thank you for showing me the value of taking initiative and connecting with other people, and for always supporting me. *Pap* – "Kijken, denken, doen." For as long as I can remember you have said this to us. Funny enough, a PhD revolves around exactly this phrase: look at a problem, think about it, and solve it! Although I have followed this advice mainly in the reversed order, thank you for teaching me this valuable lesson at such a young age, and for always being there for me.

Maurits

Amsterdam, May 2024

Contents

"I think to do something that you feel in your heart that's great, you need to make a lot of mistakes to get there. Anything that's successful is a series of mistakes."

Billie Joe Armstrong (Bullet in a Bible, 2005)

1 Introduction

In the early days of artificial intelligence (AI) research, individual sub-tasks within the field were mainly studied in isolation. For example, distinct methods and theories were developed for different tasks that depend on different data modalities, such as automatic speech recognition (e.g., Graves et al., 2006; Graves, 2012; Hannun et al., 2014), computer vision (e.g., Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016), information retrieval (e.g., Manning et al., 2008; Guo et al., 2016), knowledge representation (e.g., van Harmelen et al., 2008), and natural language processing (e.g., Mikolov et al., 2013; Sutskever et al., 2014; Bahdanau et al., 2015). It is unlikely that machines can fully comprehend a learning problem by solely using a single modality. For example, human cognition, which yields good performance on various tasks and problems (Noyes et al., 2004), can certainly not be regarded as uni-modal (Barsalou, 2001). Therefore, in this thesis, we diverge from conventional single-modal AI approaches and focus instead on multi-modal learning problems and algorithms. Multi-modal AI is defined as AI research problems that include multiple modalities of data, such as audio (speech), images/video (vision) and text (language) (Baltrusaitis et al., 2019).

There are several benefits of leveraging multiple modalities over uni-modal AI methods. For example: (i) multi-modality provides a more informative training signal since it depicts a data point from different views, thereby adding complementary information (Guo et al., 2019b). (ii) Multi-modal AI methods are capable of handling multiple modalities of data and, therefore,

enabling a more comprehensive understanding of the content and learning problem (Gautam, 2023).

Each chapter in this thesis focuses on a multi-modal learning problem. Throughout this thesis, we work with three modalities: (i) *audio*, (ii) *image(s)*, and (iii) *text*. These three modalities are studied by focusing on three multi-modal tasks: (i) *automatic speech recognition*, (ii) *scene text recognition*, and (iii) *image-caption retrieval* (or more broadly image-text representation learning). The former two tasks are characterised by their sequential nature (i.e., a sequence modeling task), while the latter is defined as a multi-modal representation learning task.

Due to the evident difference in the task characteristics that we study in this thesis, we divide the thesis into two parts. In Part 1, we focus on multi-modal *sequence modeling*. Sequence modeling tasks are characterized by the sequential nature of the input data (such as text or audio) or a model's ability to make predictions about or generate sequential output. We introduce two novel methods for multi-modal sequence modeling: one for contextual automatic speech recognition (Chapter 2) and one for scene text recognition (Chapter 3).

In Part 2, we focus on multi-modal *representation learning* for two modalities: images and text. Representation learning aims to learn representations of the input data that make it easier to extract useful information when building classifiers or other predictors (Bengio et al., 2013). The goal of image-text representation learning is to learn universal representations for both images and text, where visual concepts and textual information can be related to each other. The primary focus is on contrastive image-text representation learning, where we provide new insights into the understanding and improvement of contrastive image-text methods (Chapter 4, 5, and 6).

1.1 SCOPE AND RESEARCH QUESTIONS

This thesis does not aim to answer a single overarching research question. Instead, as stated before, it is structured in two parts, both focusing on multi-modal learning algorithms. In the first part, we focus on *multi-modal sequence modeling*. In the second part, we focus on *image-text representation learning*. Each of the five research chapters in this thesis is centered around a specific research question. We will now provide a brief overview of each research question.

We start by investigating multi-modal sequence modeling. The first task we examine is contextual automatic speech recognition. Contextual automatic speech recognition (ASR) (or contextual speech recognition) differs from conventional ASR by having access to additional user-specific context information. Alon et al. (2019) introduced a method to generate hard negative context information (i.e., phrases) for contextual ASR. However, the method by Alon et al. (2019) can be seen as a form of data augmentation before training starts and relies on an external offline ASR model to generate the hard negative phrases. In Chapter 2, which is based on (Bleeker et al., 2023a), we propose an efficient online algorithm for hard negative phrase mining: *approximate nearest neighbour phrase (ANN-P) mining*. Specifically, we formulate the following research question:

Research Question 1: *Can we improve contextual automatic speech recognition by introducing an efficient online hard negative phrase mining approach?*

The ANN-P mining method is simple to implement and can be used online during training in combination with a context-aware transformer transducer. The goal of ANN-P mining is to improve the model's ability to disambiguate between similar-sounding phrases and hence the prediction performance of the ASR model.

We demonstrate the effectiveness of ANN-P mining in a large-scale data regime consisting of 650,000 hours of acoustic training data. We show that the ANN-P mining approach results in an improvement of 7% relative word error rate reduction for the personalized portion of test data (i.e., when there is a user profile available during inference) in streaming scenarios.

The next multi-modal sequence modeling task in Part 1 is scene text recognition (STR). Similar to ASR, STR methods transcribe the text sequence present in the model input. However, STR methods take an image of text as input. To make the output transcriptions more robust, Shi et al. (2018) introduced a bidirectional STR method that decodes the text in the input image in both decoding directions: left-to-right and right-to-left. However, this comes at the cost of having two decoders – one for each decoding direction. Therefore, in the second research chapter of this thesis, we formulate the following research question:

Research Question 2: *Can we unify bidirectional multi-modal sequence modeling into a single decoder architecture for scene text recognition?*

In Chapter 3, which is based on (Bleeker and de Rijke, 2020), we propose *bidirectional scene text transformer (Bi-STET)*, a transformer-based encoder-decoder method for bidirectional scene text recognition. Due to the non-recurrent inductive bias of the transformer architecture, we can utilize the same decoder for left-to-right and right-to-left decoding. To condition the output on a decoding direction, we introduce the decoding direction embedding. We show that the Bi-STET outperforms the bidirectional STR method by Shi et al. (2018) and meets or outperforms state-of-the-art STR methods. Moreover, we demonstrate how the same attention heads are used for both right-to-left and left-to-right decoding.

In the second part of the thesis, we focus on representation learning for images and text. A prominent approach for supervised (e.g., Schroff et al., 2015; Faghri et al., 2018; Khosla et al., 2020) and self-supervised (e.g., van den Oord et al., 2018; Chen et al., 2020c) representation learning is *contrastive learning*. In contrastive learning, the goal is to learn latent representations of the input data in such a manner that similar data points are close together (i.e., a small distance given a distance metric) in the latent representation space.

In metric learning, the goal is to learn a function that maps input data into a latent space, where similar (pairs of) data are close together and dissimilar data are apart (Musgrave et al., 2020). In other words, the distance in the latent space serves as a similarity metric. To learn metric learning functions, contrastive losses are a prominent choice of optimization function and, therefore, the metric and contrastive learning fields are closely related to each other. Many contrastive learning losses have been proposed and evaluated in the context of image-to-image metric learning (e.g., Schroff et al., 2015; Movshovitz-Attias et al., 2017; Brown et al., 2020a). However, only a small number of those contrastive losses have been applied in the context of image-caption retrieval (ICR), which is an image-text representation learning task. Therefore, in Chapter 4 we raise the following research question:

Research Question 3: *Do lessons from metric learning generalize to image-caption retrieval?*

In Chapter 4, which is based on (Bleeker and de Rijke, 2022), we take several diverse metric learning functions and evaluate them in the context of the ICR evaluation task. Unexpectedly, we find that the de facto choice of loss function for the ICR task (i.e., the prominent triplet loss with semi-hard negative mining) outperforms other metric learning functions that outperform the triplet loss in different settings. To better understand why certain metric learning functions perform better than others, we introduce *counting contributing samples (COCOS)*. COCOS defines a count that tells how many samples contribute to the gradient when updating the encoder parameters for each metric learning function we examine. We show that, on average, the highest-performing loss function takes at most one negative sample into account when computing the gradient. At the same time, the underperforming contrastive losses take too many (non-informative) negative samples into account in the gradient computation.

Continuing our image-text representation learning investigation, we critically analyze one of the contrastive losses we examine in Chapter 4: the InfoNCE loss (van den Oord et al., 2018) (in Chapter 4 we refer to this loss function as the NT-Xent loss (Chen et al., 2020c)). The InfoNCE loss is a prominent optimization function for both uni- and multi-modal supervised (Radford et al., 2019; Khosla et al., 2020) as well as self-supervised representation learning tasks (van den Oord et al., 2018; Chen et al., 2020c). However, using the InfoNCE loss does not guarantee to learn all *predictive features* in the input data (Robinson et al., 2021), and this is most likely an even more prominent problem in resource-constrained training settings (i.e., when either the amount of training data or compute budget is limited). This motivates our next research question:

Research Question 4: *Can we reduce predictive feature suppression for resource-constrained contrastive image-text representation learning?*

In Chapter 5, which is based on (Bleeker et al., 2023b), we introduce *latent target decoding (LTD)*. LTD is an additional decoding objective that we add to the contrastive ICR framework, which reconstructs the input caption in a latent space of a general-purpose sentence encoder in a non-auto-regressive

manner. Instead of implementing LTD as a dual optimization objective (i.e., an additional loss function), we propose to implement LTD as an optimization constraint. We demonstrate that LTD reduces predictive feature suppression by obtaining higher recall@k, r-precision, and nDCG scores than baseline ICR methods that are optimized using a contrastive loss. Additionally, we show that LTD can be applied with different contrastive losses and ICR methods. Furthermore, we find that implementing LTD mainly reduces predictive feature suppression when implemented as an optimization constraint, rather as a dual optimization objective.

If a image-text encoder model minimizes a contrastive loss while still suppressing predictive features in the input data, one could argue that the model relies on a *shortcut* to optimize the objective (Robinson et al., 2021), failing to capture all task-relevant information for the given task. In Chapter 5, we show that using LTD for contrastive image-text methods can mitigate predictive feature suppression (measured by the generalizability of the learned representations to the ICR evaluation task). However, it remains unclear to what extent contrastive image-text methods rely on shortcut solutions when minimising the InfoNCE objective. Therefore, we formulate the final research question as follows:

Research Question 5: *Can we demonstrate and reduce shortcuts in contrastive image-text representation learning?*

In Chapter 6, which is based on (Bleeker et al., 2024), we refer to image-text with the broader term vision-language. We introduce a *synthetic shortcuts for vision-language (SVL)* framework. SVL is a training and evaluation framework that allows for the injection of synthetic shortcuts into image-text data in a controlled manner. The SVL framework allows us to measure how much of the task-relevant information in the input (i.e., predictive features) is still captured by contrastive image-text methods when a shortcut is present in the training data. We show that contrastive image-text methods that are either trained from scratch (VSE++, Faghri et al., 2018) or fine-tuned (CLIP, Radford et al., 2021) with data containing synthetic shortcuts, mainly learn features that represent the shortcut and that the remaining task-relevant information is suppressed. Therefore, we conclude that contrastive losses are not sufficient to learn task-optimal representations that contain all predictive features in the input data.

Next, we examine two shortcut reduction methods on the SVL framework: (i) latent target decoding (Bleeker et al., 2023b), and (ii) implicit feature modification (Robinson et al., 2021). We find that both methods improve performance on the ICR evaluation task (when shortcuts are included in the training data) in some settings, however, they only partially reduce shortcut learning when training and evaluating with the SVL framework.

This concludes the overview of the research questions we answer in this thesis. In the next section, we provide an overview of the contributions of this thesis.

1.2 MAIN CONTRIBUTIONS

Algorithmic contributions

- We propose *approximate nearest neighbour phrase* (ANN-P) mining, a novel online hard negative mining method combined with a context-aware transformer transducer for contextual speech recognition (Bleeker et al., 2023a, Chapter 2).
- We propose *bidirectional scene text transformer* (Bi-STET), a novel method for bidirectional scene text recognition with a single decoder network (Bleeker and de Rijke, 2020, Chapter 3).
- We propose *counting contributing samples* (COCOS), a novel method that makes it possible to compare contrastive loss functions by counting how many samples contribute to the gradient w.r.t. the query representation (Bleeker and de Rijke, 2022, Chapter 4).
- We propose *latent target decoding* (LTD), a novel method to reduce predictive feature suppression for ICR methods in resource-constrained scenarios (Bleeker et al., 2023b, Chapter 5).
- We propose the *synthetic shortcuts for vision-language* (SVL) framework, a novel training and evaluation framework that allows us to inject synthetic shortcuts into image-text data to measure to what extent contrastive image-text methods rely on shortcuts to minimize the contrastive optimization objective (Bleeker et al., 2024, Chapter 6).

Theoretical contributions

- We show that contrastive losses (i.e., InfoNCE (van den Oord et al., 2018)) that enforce minimal sufficient representations can never learn task-optimal image representations (i.e., representations that contain all task-relevant information in the input captions), in the context of image-text representation learning with multiple matching captions per image (Bleeker et al., 2024, Chapter 6).

Empirical contributions

- We show that ANN-P mining results in a 7% relative word error rate reduction on the personalized fraction of the test data (i.e., when a user information or profile is available) compared to a context-aware transformer transducer that is trained with random phrase mining in streaming scenarios (Bleeker et al., 2023a, Chapter 2).
- We show that Bi-STET outperforms the bidirectional STR method by (Shi et al., 2018), and either outperforms or meets the performance of state-of-the-art (SOTA) STR methods (Bleeker and de Rijke, 2020, Chapter 3).
- We empirically evaluate if findings from metric learning generalize to the ICR task (Bleeker and de Rijke, 2022, Chapter 4).
- We show that constrained-based LTD reduces predictive feature suppression, as it outperforms ICR baselines that are only optimized by using a contrastive loss or LTD (Bleeker et al., 2023b, Chapter 5).
- We compare constraint-based LTD with LTD implemented as a dual optimization objective. We show that LTD implemented as a dual loss is less effective to reduce predictive feature suppression (Bleeker et al., 2023b, Chapter 5).
- We evaluate the SVL framework in a wide range of different settings, varying the amount of shortcuts we add to the training and evaluation data. We show that, in general, the more shortcuts we add to the training data, the more the contrastive image-text methods learn to represent shortcut features in the input data, suppressing remaining task-relevant information (Bleeker et al., 2024, Chapter 6).
- We examine two shortcut reduction methods (latent target decoding and implicit feature modification) on the SVL framework and show that shortcut learning partially can be mitigated in some settings (Bleeker et al., 2024, Chapter 6).

1.3 THESIS OVERVIEW

This section provides an overview of this thesis and some recommendations for reading directions. This thesis consists of seven chapters: an introduction, five research chapters and a conclusion. Each research chapter is based on a publication and is centered around a specific research question (as discussed in Section 1.1). The research chapters are self-contained and can be read independently and in any desired reading order.

However, each chapter in Part 2 is a direct follow-up of the preceding one, following a chronological order of publication. Therefore, Chapter 4, 5, and 6 are best read together in order, since they build on each other. Although each chapter in Part 2 is a direct follow-up of the preceding one, each chapter is still based on an individual publication. As a result, the notation and terminology between the chapters in Part 2 can differ somewhat. To avoid confusion in the used notation, we provide a notation table in the appendix of Chapter 4, 5, 6. We conclude this thesis by providing a summary of our findings and outlining future research directions in Chapter 7.

1.4 ORIGINS

Each chapter in this thesis is based on a publication. Below we list the publications that are the origins of each chapter.

- **Chapter 2**

  Maurits Bleeker, Pawel Swietojanski, Stefan Braun, and Xiaodan Zhuang. Approximate Nearest Neighbour Phrase Mining for Contextual Speech Recognition. In *INTERSPEECH'23*. 2023.

  This work was done during an internship at Apple AI/ML in 2022. MB scoped the initial idea. MB implemented the algorithm, based on discussions with PS, and ran the initial experiments. PS further ran the final experiments and had an important advisory role. SB provided the pre-trained BERT model for the context encoder. All authors contributed to the writing. MB and PS contributed equally to the writing.

- **Chapter 3**

  Maurits Bleeker and Maarten de Rijke. Bidirectional Scene Text Recognition with a Single Decoder. In *ECAI'20*. 2020.

  MB proposed the idea, designed and ran the experiments. MdR had an important advisory role. All authors contributed to the writing. MB did most of the writing.

- **Chapter 4**

  Maurits Bleeker and Maarten de Rijke. Do Lessons from Metric Learning Generalize to Image-Caption Retrieval? In *ECIR'22*. 2022.

  MdR proposed the idea of doing a reproducibility study. MB proposed the experimental setup, designed and ran the experiments. MdR had an important advisory role. Both authors contributed to the writing. MB did most of the writing.

- **Chapter 5**

  Maurits Bleeker, Andrew Yates, and Maarten de Rijke. Reducing Predictive Feature Suppression in Resource-Constrained Contrastive Image-Caption Retrieval. In *TMLR*. 2023.

  MB proposed the idea, designed and ran the experiments. MdR and AY had important advisory roles. All authors contributed to the writing. MB did most of the writing.

- **Chapter 6**

  Maurits Bleeker, Mariya Hendriksen, Andrew Yates, and Maarten de Rijke. Demonstrating and Reducing Shortcuts in Vision-Language Representation Learning. *Under review*. 2024.

  MB proposed the idea, designed the shortcut sampling framework, and the shortcut reduction experiments. MB ran the shortcut sampling experiments for CLIP, and MH for VSE++. MB ran all the shortcut reduction experiments. MdR and AY had important advisory roles. All authors contributed to the writing. MB and MH contributed equally to the writing.

The writing of this thesis also benefited from work on the following publications:

- David Stap, **Maurits Bleeker**, Sarah Ibrahimi, and Maartje ter Hoeve. Conditional Image Generation and Manipulation for User-Specified Content. In *CVPR, AI for Content Creation Workshop*. 2020.
- Michael Neely, Stefan F. Schouten, **Maurits Bleeker**, and Ana Lucic. Order in the Court: Explainable AI Methods Prone to Disagreement. In *ICML, Theoretic Foundation, Criticism, and Application Trend of Explainable AI Workshop*. 2021.
- Mariya Hendriksen, **Maurits Bleeker**, Svitlana Vakulenko, Nanne van Noord, Ernst Kuiper, and Maarten de Rijke. Extending CLIP for Category-to-image Retrieval in E-commerce. In *ECIR'22*. 2022.
- Ana Lucic, **Maurits Bleeker**, Sami Jullien, Samarth Bhargav, and Maarten de Rijke. Reproducibility as a Mechanism for Teaching Fairness, Accountability, Confidentiality, and Transparency in Artificial Intelligence. In *AAAI Symposium on Educational Advances in Artificial Intelligence*. 2022.
- Ana Lucic, **Maurits Bleeker**, Samarth Bhargav, Jessica Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, and Robert Stojnic. Towards reproducible machine learning research in natural language processing. *ACL Tutorial on Reproducibility*. 2022.
- Michael Neely, Stefan Schouten, **Maurits Bleeker**, and Ana Lucic. A Song of (Dis)agreement: Evaluating the Evaluation of Explainable Artificial Intelligence in Natural Language Processing. In *HHAI'22*. 2022.
- Ana Lucic, **Maurits Bleeker**, Maarten de Rijke, Koustuv Sinha, Sami Jullien, and Robert Stojnic. Towards Reproducible Machine Learning Research in Information Retrieval. *SIGIR Tutorial on Reproducibility*. 2022.
- **Maurits Bleeker**. Multi-modal Learning Algorithms and Network Architectures for Information Extraction and Retrieval. In *ACM MM'22*. 2022.

Part 1 - Multi-modal Sequence Modeling

2

Phrase Mining for Contextual Speech Recognition

We start this thesis by examining the first multi-modal task: automatic speech recognition (ASR). ASR is a sequential multi-modal task and the objective of ASR methods is to transcribe the spoken language in the audio into text.

In this chapter,[1] we specifically focus on *contextual* speech recognition. In contextual speech recognition, the ASR model has access to additional context information to bias the output prediction towards user-specific utterances. This can be done by providing the ASR model a *context list* containing user-specific context phrases (e.g., Pundak et al., 2018; Chang et al., 2021).

To make the predictions of the ASR model more robust to similar-sounding phrases, Alon et al. (2019) introduced a hard negative mining approach for contextual speech recognition. However, this method runs offline (i.e., before training the contextual ASR method) by using an external ASR method. Therefore, in this chapter, we raise the first research question of this thesis:

Research Question 1: *Can we improve contextual automatic speech recognition by introducing an efficient online hard negative phrase mining approach?*

To answer this research question, we introduce *approximate nearest neighbour phrase (ANN-P) mining* for contextual speech recognition. ANN-P mining is a novel method that samples hard negative phrases online during training from the latent space of the context encoder. We show that our ANN-P mining method results in up to 7% relative word error rate reductions for the personalized portion of test data in streaming scenarios, with only minor regressions on generic queries.

1 This chapter is based on (Bleeker et al., 2023a).

2.1 INTRODUCTION

Recognizing words that are rare or unseen during training poses a challenge for end-to-end (E2E) automatic speech recognition (ASR) (Sainath et al., 2018; Alon et al., 2019; Bruguier et al., 2019; Guo et al., 2019a). One way to address this problem is to allow the model to use user-specific information during inference, such as contact names, app names, media titles, and relevant geo-location names. To that end, several approaches have been proposed including shallow language model (LM) fusion (Zhao et al., 2019; Le et al., 2021), on-the-fly rescoring (Hall et al., 2015; Williams et al., 2018; Zhao et al., 2019), or deep fusion approaches (Pundak et al., 2018; Bruguier et al., 2019; Jain et al., 2020). Since E2E models tend to learn a strong internal LM (McDermott et al., 2021; Meng et al., 2021), shallow LM fusion and rescoring approaches are not always effective out of the box.

Alternative methods rely on deep neural contextual fusion (DCF) (Pundak et al., 2018; Bruguier et al., 2019; Jain et al., 2020; Chang et al., 2021; Sun et al., 2021; Munkhdalai et al., 2022; Sathyendra et al., 2022). In DCF, the biasing machinery is part of the ASR model and is jointly learned with the main ASR objective. Different DCF techniques share much of the same modeling back-end and thus can be implemented for arbitrary E2E network architectures such as the attention encoder-decoder (AED) (Chan et al., 2015) or the RNN Transducer (RNN-T) (Graves, 2012) ASR systems. Deep contextual biasing has been proposed for the contextual listen, attend, and spell (CLAS) (Chan et al., 2015; Pundak et al., 2018) AED model and similar solutions were extended to the contextual neural transducer (Jain et al., 2020; Chang et al., 2021). The major difference between contextual models and their non-contextual counterparts is the biasing machinery, usually implemented as an additional context encoder followed by a fusion mechanism. The context encoder is typically implemented as an LSTM (Hochreiter and Schmidhuber, 1997) or more recently a transformer (Vaswani et al., 2017) model, and its role is to project a set of tokenized biasing phrases into a set of fixed-sized continuous embeddings. Next, a fusion mechanism integrates these embeddings with the acoustic (AED, RNN-T) and/or label (RNN-T) encoder when making ASR predictions. Fusion can be implemented in a latent space with cross-attention between audio and context encoders (Pundak et al., 2018; Jain et al., 2020) or by interpolat-

ing generic and contextual model's distributions, as done in tree-constrained pointer generation networks (TCPGN) (Sun et al., 2021).

A major challenge in contextual biasing is that some words, including the biasing phrases fed into the context encoder, may exhibit phonetic similarities with one another or may be characterized by complex and non-standard pronunciation patterns. For example, names in a contact list that sound similar to each other, or geo-location names that have similar (but not identical) pronunciations. To make deep contextual biasing more robust to settings where context information is (phonetically) similar to each other, one could explicitly embed additional phoneme-level information in the contextual ASR as explored in (Chen et al., 2019b), or train the ASR system such that it can learn to better disambiguate between challenging queries.

In this chapter, we are interested in the latter approach by exposing the ASR to hard negative examples during training. Alon et al. (2019) proposed a method to generate phonetically similar phrases given a reference phrase. Phonetically similar phrases might have similar (i.e., confusing) acoustic representations and, therefore, are hard to distinguish from the desired phrase during inference (Alon et al., 2019). By appending phonetically similar phrases as *hard negatives* to the context encoder's inputs during training, the model is explicitly tasked to disambiguate between them. There are several possible ways to insert hard negatives into the training pipeline. In Alon et al. (2019), an external ASR model (Variani et al., 2017) is used to decode and generate a set of hypotheses for each query. These hypotheses are then ranked based on the word co-occurrence and the phonetic similarity with the reference phrase.

While the method by Alon et al. (2019) has shown promising results, it is worth noting that their approach may be viewed as a form of data augmentation implemented prior to training of a deep contextual ASR model, rather than a technique that can be directly integrated into the training process. This may not be optimal, as exact hard negative phrases (HNP) are likely to depend on the mistakes a specific ASR is prone to make, rather than mis-recognitions of some independent ASR system.

In this chapter, we present an alternative, computationally efficient extension of mining HNP: approximate nearest neighbour phrase (ANN-P) mining. ANN-P mining allows one to efficiently select HNP during the training of a deep contextual model in an online manner, using the latent space of the context encoder. ANN-P mining unifies two important aspects of HNP for contex-

tual ASR in a single method, by including phrases in the context list that are (i) phonetically similar to the reference phrase (i.e., distracting phrases), as in (Alon et al., 2019), and (ii) close to the reference phrase in the latent space of the context encoder. The latter property is hypothesized to make it harder for the model to discriminate different phrases (Alon et al., 2019). Different from Alon et al. (2019), our approach does not require a full decode of training data, nor an existing pre-trained ASR model to obtain hard negatives.

We implement the proposed ANN-P mining using the context-aware transformer transducer (CATT) model (Chang et al., 2021). CATT proposed an efficient biasing approach that makes full use of the transformer transducer (TT) (Zhang et al., 2020) architecture. The biasing phrases in the CATT are mapped into single embeddings (i.e., one per biasing phrase) that are then used with cross-attention to bias both the audio and label encoders. While the original CATT formulation only considered non-streaming scenarios, we extend CATT to both streaming and non-streaming applications at the same time by training it in a variable attention masking manner (Tripathi et al., 2020; Swietojanski et al., 2023). Given the limited (audio) context window in streaming scenarios, it is difficult to accurately distinguish the correct biasing phrase while transcribing. This especially applies for CATT-like approaches where each biasing phrase is compressed into a single embedding, rather than an embedding per sub-word as is the case with TCPGN or neural associative networks (NAM) (Munkhdalai et al., 2022).[2] As such, we anticipate that for the CATT model, ANN-P mining is likely to provide greater value in streaming than in non-streaming scenarios.

To summarize, the contributions of this chapter include:

- An extension of the training mechanism for the CATT that utilizes mining hard negative phrases directly from the latent space of the ASR model. This approach results in up to 7% relative word error rate (WER) reductions for the personalized portion of test data, with only minor regressions on generic queries or around 2% relative on average when taking into account both generic and personal queries.
- An extension of the CATT to the streaming scenario.

2 Although not investigated in this chapter, CATT is likely to work well for neural biasing of non-auto-regressive models like connectionist temporal classification (CTC) (Graves et al., 2006) whereas NAM or TCPGN relies on decoded prefixes for biasing. See (Dingliwal et al., 2023) for a recent CTC study.

- Evaluation of the proposed approach in a large-scale data regime consisting of 650,000 hours of acoustic training data. This is to make sure the models are well-trained, and not handicapped by limited training data diversity, which in turn could artificially inflate biasing performance.

2.2 METHOD

2.2.1 Context-aware transformer transducer

The context-aware transformer transducer (Chang et al., 2021) extends an RNN-T consisting of a label and audio encoder, with an additional contextual encoder, followed by a biasing cross-attention layer that measures the relevance of the context phrase to the query from the perspective of information found in the audio and label encoders. The context encoder $f^{context}(\cdot)$ takes as input a set of context phrases $\mathcal{S}_\mathcal{C} = \{w_1, \dots\}$ and maps each context phrase $w_i \in \mathcal{S}_\mathcal{C}$ into a fixed vector representation $f^{context}(w_i) := \mathbf{h}_i^{CE}$. The CATT uses a transformer (Vaswani et al., 2017) for the label, audio, and context encoder.

The biasing layer in the CATT model consists of a multi-head attention (MHA) layer (Vaswani et al., 2017). The goal of the MHA is to measure the similarity between each phrase in the context list and the audio signal using scaled dot-product cross-attention, which weights each input phrase according to this similarity score with the audio.

Consider a pair of contextual phrases $w_i, w_j \in \mathcal{S}_\mathcal{C}$, where $i \neq j$. Phrase w_i is the reference phrase in the audio signal and w_j is considered as a negative phrase w.r.t. the audio transcript. Phrases i and j have close to each other (key) embeddings ($\mathbf{k}_i$ and $\mathbf{k}_j$ respectively), in terms of a similarity metric. When a query-key pair has a low attention score, the matching value embeddings get a close to zero weight score. Since $\mathbf{k}_i$ and $\mathbf{k}_j$ are close to each other, $\mathbf{k}_j$ may be distracting for the attention head(s), yielding a too high attention score, hence the distracting phrase may be taken into account when transcribing the audio.

2.2.2 Proposed method: ANN-P mining

In this section, we present an approach to extend the training of the CATT by introducing ANN-P mining. The goal of ANN-P mining is to select phrases

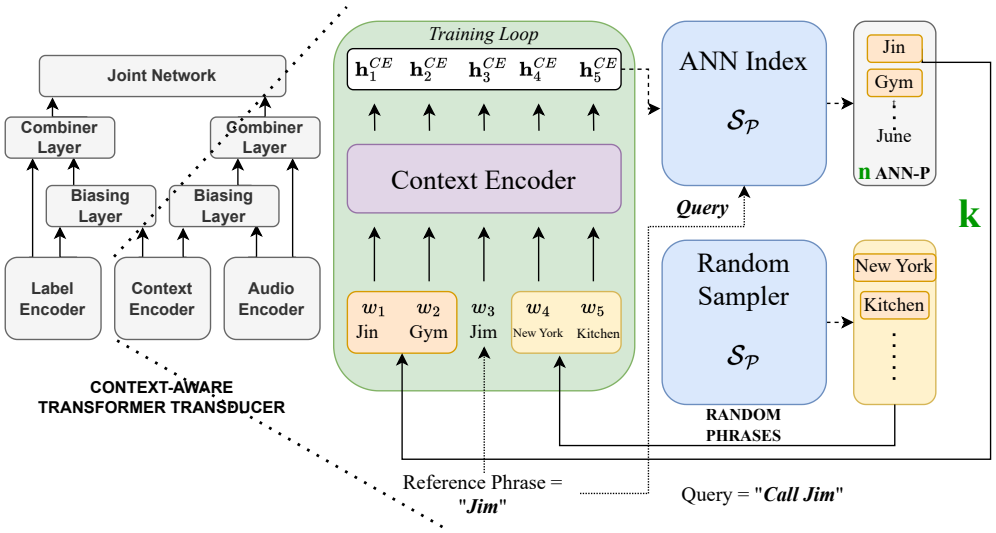


Figure 2.1: Overview of the ANN-P mining method. Given a query **"Call Jim"**, we append k ANN phrases to the context list. Given a query phrase, we first sample n phrases from the ANN index. Next, we sample the top k. The remaining phrases in the context list are randomly sampled.

similar to the *reference phrase* in terms of their similarity in the latent space of the context encoder. When mining the phrases randomly, the probability of having (phonetically) similar phrases in the context list is negligible. Therefore, the label, audio, and context encoder may not learn to disambiguate between similar sounding (i.e., difficult to discriminate) phrases. In Figure 2.1 we provide a high-level overview of our ANN-P mining method.

Prior to training, we extract all the biasing phrases from each audio transcription using the existing automatically generated meta-information on entity spans. This results in a set of phrases for the entire training data $\mathcal{S}_\mathcal{P}$, referred to as the biasing phrases inventory. $\mathcal{S}_\mathcal{P}$ can be extended with additional entries that are not present in the training data to provide additional context, as needed.

The goal of ANN-P mining is to select hard negative samples according to the current state of the trained ASR model. We use the context encoder of the CATT to encode each phrase $w_i \in \mathcal{S}_\mathcal{P}$ into its latent representation $\mathbf{h}_i^{CE}$ given a checkpoint of the CATT during training (left box in Figure 2.1). We cache the latent representation $\mathbf{h}_i^{CE}$ of each phrase into an online approximate nearest neighbour (ANN) index (i.e., an index that can be efficiently queried during training).

Given a *query phrase* w_i, the ANN index maps the phrase w_i to its cached latent representation $w_i \rightarrow \mathbf{h}_i^{CE}$ and, using ANN search over all cached phrase embeddings, returns n ANN-Ps from the index based on the dot product score w.r.t. the query phrase. To prevent sampling the same phrases at every epoch for the same query, we randomly sample k (where $k < n$) phrases from the n retrieved phrases. The remaining phrases (if any) that are needed for the given query are added randomly by sampling from $\mathcal{S}_\mathcal{P}$. Together the ANN-Ps and the randomly sampled phrases are included in the context list. ANN-P mining can only be used if the query phrase is present in the ANN index since we need its neural representation to apply similarity search. Hence, we need to cache each phrase first before we can apply ANN-P mining. However, an indexing step is an inexpensive process, taking a small percentage of the total training time.

For ANN-P mining, we need one representation per biasing phrase (all from the same latent space) to store in the ANN index. In this chapter, we use the output representation of the context encoder $\mathbf{h}_i^{CE}$ for ANN-P mining. However, CATT measures the similarity between the audio and the phrases by using a MHA biasing layer, which uses 8 attention heads. Hence, there are 8 different phrase representations (living in different latent spaces) that are used to compute an attention score (i.e., similarity) between a phrase and the audio signal. There are several ways to aggregate 8 key representations into a single representation that could be used for approximate nearest neighbour search. As a straightforward approach that applies as an approximation for all the 8 key representations, we store the output representation of the context encoder $\mathbf{h}_i^{CE}$ in the index instead (this is because the key projection is only a linear transformation).

2.3 EXPERIMENTAL SETUP

We carry out the experiments on a large-scale dataset consisting of queries from two tasks: dictation and assistant. The semi-supervised portion of the data consists of around 600,000 hours of randomized and anonymized automatically transcribed acoustic data, while the supervised part contains around 50,000 hours of randomized and anonymized English queries.

Following (Nguyen et al., 2022; Swietojanski et al., 2023), our systems are trained in a two-stage manner – the first stage pre-trains the models on semi-supervised data for a total of 5.6 million updates, the second fine-tunes the model for another 280 thousand updates on supervised data. In both stages, the gradients are accumulated over 9216 queries. We use SyncSGD + Adam (Kingma and Ba, 2015) for distributed optimization, with exponentially decaying learning rates. ANN-P sampling is applied only in the fine-tuning stage and not during evaluation. All models are evaluated using a test set containing 60 hours of assistant data. Around 40% of the test set consists of contextual queries spanning domains such as contact, app, and geo-location names. During inference, we include real user profiles in the context list. The remainder of the test set consists of queries that are generic in nature and are unlikely to benefit from personalized priors.

2.3.1 Contextual transformer transducer model

In this chapter, our base contextual E2E ASR model is the context-aware transformer transducer (CATT) (Chang et al., 2021), configured to have around 120 million parameters. The audio encoder is a 12-layer Conformer (Gulati et al., 2020) while the label and context encoders are implemented as a 6-layer transformer model. Each encoder has an embedding size of 512 and the MHA is configured to 8 attention heads. All examples in the training batch share the same context list, which allows exposing each query to a larger number of context phrases while keeping memory usage low. To do so, we only sample a few random + ANN-Ps per query and combine them into a single context list for all queries in the mini-batch. This was configured such that each query has access to around 96–128 biasing phrases.

Different from the original CATT, we append a back-off phrase to the context list that the model can attend to in case there are no relevant biasing phrases. Adding a back-off token has also been demonstrated to be effective with the CLAS (Pundak et al., 2018). Originally, the CATT was only trained and evaluated with global context models. In this chapter, we also investigate its suitability to both non-streaming and streaming applications. We do so by training both CATT and baseline models in a variable masking manner (Tripathi et al., 2020; Swietojanski et al., 2023), and then configuring the models to either streaming or non-streaming settings during decoding. Streaming mod-

els operate on 240ms long causal audio chunks (Chen et al., 2021b; Shi et al., 2021) and thus have limited access to the future audio signal which may pose a challenge for CATT approach.

To show performances without the biasing machinery, we compare with a transformer transducer (TT) (Zhang et al., 2020) that has the same audio and label encoder architecture as CATT and has been trained with multiple attention masks to allow for streaming and non-streaming decoding. The exact architecture details of the TT can be found in (Swietojanski et al., 2023). Since streaming models are expected to emit tokens with low partial latency, we train both TT and CATT models with the latency-penalizing FastEmit loss (Yu et al., 2021a).

2.3.2 ANN index and negative phrase mining

The ANN-P index is built using the Annoy[3] library. For the ANN-P mining, we experiment with various ways of mixing negative examples into context lists. In general, we append 8 biasing phrases for each query,[4] where some proportion is expected to be made of ANN-Ps (see Section 2.4.2 for details), while the remainder is randomly selected out of the biasing phrases inventory. We use the dot product between phrase embeddings and a query as similarity metric to mine ANN-Ps.

Given a training query, and its corresponding reference biasing phrase(s) (if any), we retrieve n approximate nearest neighbours from the index. Important to notice, we sample each negative phrase at the word level (i.e., if a query phrase consists of multiple words, we sample n HNPs per word). Next, we sample k phrases at random from the retrieved n ANNs. After completing two fine-tuning epochs, the ANN index is rebuilt by re-indexing the phrase representations using the latest state of context encoder parameters. To prevent over-fitting on the same ANN-Ps, we do not sample ANN-Ps for every query in the training batch but use them proportionally to the *append ratio*. The frequency of adding ANN-Ps to the context list during training increases as

3 `https://github.com/spotify/annoy`

4 Note, we eventually share these across all examples in the batch, so for a batch of 16 queries and 8 contextual phrases per query, each query would make use of 128 biasing phrases to pick from.

the append ratio increases. In cases where we are unable to sample ANN-Ps for a query, we use random phrases instead.

2.4 RESULTS

2.4.1 Random vs. ANN-sampled phrases

Table 2.1 reports the results for models operating in global (i.e., non-streaming) (upper block) and streaming (lower block) modes, with and without access to contextual information. To demonstrate the effect on the WER in situations where contextual information is absent, we also evaluate the CATT without access to relevant contextual information. In this chapter, global and streaming models are the same models, configured to different operating regimes via different settings of attention masking (Swietojanski et al., 2023). Note that Chang et al. (2021) only investigated the biasing performance of CATT in a global setting, and thus it is unclear if and to what extent the CATT approach can be used in streaming applications.

When decoding CATT models in global mode, allowing the model to access contextual information during inference improves accuracy by 38% relative WER (WERR) on average (i.e., 6.5% vs. 3.9% for non-biased and biased CATT systems, respectively). This is accompanied by a 3% WERR degradation on the non-personal (generic) portion of the test set (i.e., 6.3% vs. 6.5% on average for TT and CATT, respectively). These results are in line with findings on CATT and global decodes reported by Chang et al. (2021). Another observation is that training with ANN-P mining does not seem to affect the non-streaming results in a significant way. This can be most likely explained by the fact that having access to the entire audio sequence allows the model to better contextualize the information, thus it is easier to match the complete audio evidence to the correct biasing phrase.

When decoding CATT models configured to streaming mode, we observe similar overall trends as with the global models. Interestingly, for the streaming scenario, the regression on the WER for the non-contextual portion of data no longer exists when compared to a baseline TT model (i.e., 6.8% vs. 6.7% for the baseline and CATT, respectively). For the streaming scenario, we also

Table 2.1: Word error rate (WER) for models configured to global (upper block) and streaming (lower block) decodings, with and without access to contextual information (Ctx. Info). Context-aware transformer transducer (CATT) models are trained with random, or approximate nearest neighbour phrase (ANN-P) sampling. The results on test data are additionally aggregated on generic and personalized subsets. The Generic portion consists of queries that are not expected to benefit from contextual information. Results obtained for $n = 20$, $k = 2$, and append $ratio = 0.25$.

Model	Ctx. Info	WER [%]		
		Generic	Personal	Avg.
Global decoding				
TT	-	3.9	11.9	6.3
CATT	✗	4.3	11.5	6.5
CATT w/ ANN-P	✗	4.4	11.5	6.5
CATT	✓	4.5	2.6	3.9
CATT w/ ANN-P	✓	4.5	2.6	3.9
Streaming decoding				
TT	-	4.5	12.3	6.8
CATT	✗	4.6	11.4	6.7
CATT w/ ANN-P	✗	4.5	11.6	6.7
CATT	✓	4.9	2.8	4.3
CATT w/ ANN-P	✓	4.9	**2.6**	**4.2**

obtain up to 7% WERR reductions on the contextual portion of the test set (this is where ANN-P mining is expected to help).

We conclude that for the limited audio look-ahead, training CATT with ANN-P mining helps to improve accuracy by allowing the model to better disambiguate between contextual phrases. Since in CATT the biasing information is compressed into a single embedding, the use of HNP helps to regularize embeddings such they are more robust to small phonetic variations.

2.4.2 Further analyzes

ANN-P mining has three main hyper-parameters: n (i.e, the number of ANN-Ps we take from the index), k (i.e., the k phrases we sample from the n ANNs), and the *append ratio* (how frequent we add the ANN-Ps to the con-

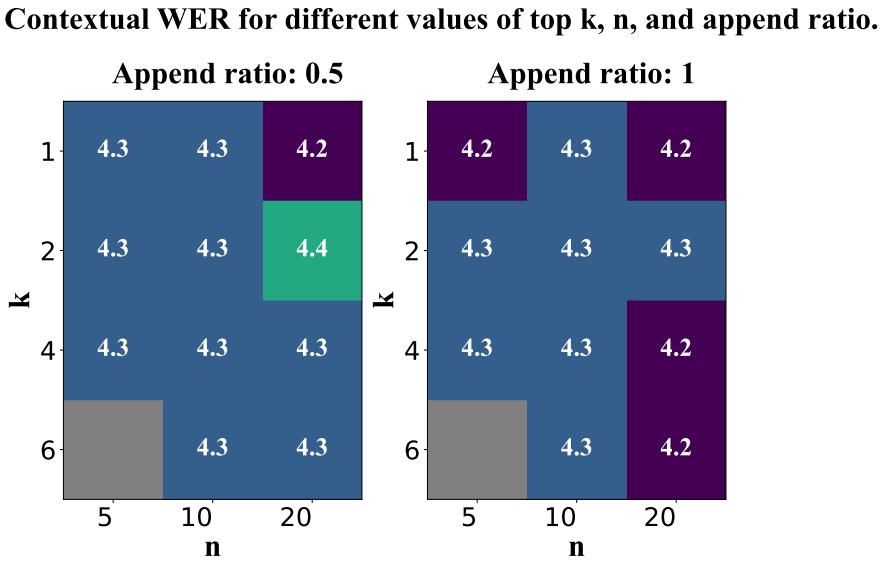


Figure 2.2: Average word error rate (WER) for streaming decodes using different values of top k, n, and append ratio.

Table 2.2: Query phrases and their ANNs retrieved from the index.

Query phrase	$n = 4$ - ANN Phrases
john	'joan', 'johnson', 'johann', 'from john'
building	'buildings', 'builder', 'the building', 'builds'
jean	'jeanne', 'jeannie', 'jeana', 'jeanine'
eva	'evie', 'ava', 'evin', 'evy'
play	'playa', 'place', 'flay', 'platte'

text list). We depict the effect of each parameter in Figure 2.2. We can conclude that, in general, ANN-P mining is robust to the choice of the considered hyper-parameters. The lowest scores are obtained for an append-ratio of 1 and using $n = 20$ phrases from the index. The number of phrases (k) appended to the context list does not seem to have a strong effect on the WER.

In this chapter, we mine hard negative phrases (at the word level) from the latent space of the context encoder, based on the neural similarity with the reference (i.e., query) phrase. In Table 2.2, we provide several examples of query phrases and their top four ANN-Ps as retrieved from the index. We can observe that top ANN-Ps mainly results in phrases that are phonetically similar to the query phrase. For queries consisting of names, we mainly retrieve other similar-sounding (but different) names. For a query such as *building*, we mainly

retrieve ANN-Ps that are related to the same concept and contain the sub-word *build*.

Finally, we also investigated the following aspects and report them here for completeness. These experiments either did not significantly impact accuracy or led to deterioration:

- Rebuilding the ANN-P index at different epochs did not have a significant impact on the WER.
- Enabling ANN-P mining at different stages of training, including the pre-training, or fine-tuning the last 2–3 epochs did not improve over using it during the entire fine-tuning stage.
- Sampling ANN-Ps using multi-word phrases, instead of single-word phrases, resulted in 5% WERR degradation.

2.5 DISCUSSION & CONCLUSION

In this chapter, we answered the first research question of this thesis positively by proposing and evaluating an efficient online method for mining approximate nearest neighbour phrases (i.e., hard negatives) for transformer-transducer contextual speech recognition based on CATT model: approximate nearest neighbour phrase mining. In order to mine hard negatives, our method does not require an offline external ASR model, nor additional decodings of training data. We also extended CATT modeling to streaming applications by training it with multiple attention mask configurations. We evaluated the proposed ideas in large-scale data experiments, finding that the CATT using the ANN-P mining approach offers up to 7% relative WER reductions for streaming models on the personalized portion of the test data. Hence, we conclude that ANN-P mining improves contextual ASR.

In the next chapter, we continue with multi-modal sequence modeling, but with a different task: scene text recognition. The modeling principles of scene text recognition (STR) align closely with those of ASR. Both ASR and STR transcribe the text that is present in the input data. However, instead of audio features, STR methods take images as input.

3

Bidirectional Scene Text Recognition

In this chapter, we turn our focus to a second multi-modal sequence modeling task: scene text recognition (STR). STR methods take an image of a sequence of characters (or words) as input and strive to decode the characters in the input image. The standard modeling paradigm uses a convolutional neural network (CNN) as a feature extractor, and a recurrent neural network (RNN) to decode the character sequence in the input image. To improve the robustness of STR methods, a bidirectional decoding method has been introduced (Shi et al., 2018). However, this comes with the cost of having two decoders, one for each decoding direction. In this chapter,[1] we answer the second research question of this thesis:

Research Question 2: *Can we unify bidirectional multi-modal sequence modeling into a single decoder architecture for scene text recognition?*

To answer this research question, we propose the *bidirectional scene text transformer (Bi-STET)*. Bi-STET is a transformer-based encoder-decoder method. Due to the non-recurrent inductive bias of the transformer, we can utilise the same decoder network for both left-to-right and right-to-left decoding. We show that Bi-STET achieves or outperforms state-of-the-art STR methods with a simpler approach than other bidirectional STR methods

3.1 INTRODUCTION

Scene text recognition (STR) is the task of recognizing the correct word or character sequence in a cropped word image. Many different architectures have been proposed for STR. Since the rise of deep learning, most state-of-the-art

1 This chapter is based on (Bleeker and de Rijke, 2020).

STR methods adopt a convolutional neural network (CNN) for feature extraction and an encoder-decoder architecture as the core component for sequence modeling. After feature extraction with a CNN, the extracted features of the input image are encoded into a new representation with an encoder. As a final step, conditioned on the encoded input image representation, the character sequence is decoded, which is depicted in the input image. Figure 3.1 summarizes a general pipeline. Baek et al. (2019) identify sequence modeling as a core component in STR frameworks.

The encoder-decoder architecture for the sequence modeling stage of many state-of-the-art STR methods (see Section 3.2) can be characterized by (i) a bidirectional RNN for feature encoding, (ii) a directed RNN decoder for character decoding, and (iii) various types of attention mechanism (Bahdanau et al., 2015; Luong et al., 2015) to generate additional context vectors for the current decoding steps.

Two recent developments have accelerated progress in STR: (i) a move away from recurrent sequence modeling, and (ii) bidirectional decoding for STR.

Regarding the first, Sheng et al. (2018) have changed the standard STR approach for sequence modeling by introducing a non-recurrent method based on a transformer encoder-decoder architecture (Vaswani et al., 2017). By using a transformer-based encoder-decoder, the model architecture can be simplified and the time for model optimization can be reduced by an order of magnitude in comparison (Sheng et al., 2018).

The second reason for recent progress in STR is bidirectional decoding. Bidirectional decoding is the idea of decoding an output sequence in two directions (i.e., from left-to-right and right-to-left) for more robust output predictions. This bidirectional decoding is implemented by using a different decoder for each decoding direction (Shi et al., 2018). Decoding the text in two directions at the same time can be seen as two different sub-tasks for the model to perform.

It is important to reflect on different ways of modeling sub-tasks, especially using task conditioning. With task conditioning, the output of a method does not solely depend on the input data, but on a given (sub-)task as well. In other words, given the same input data, the output may be different based on the (sub-)task it is conditioned on. As explained by Radford et al. (2019), task conditioning can be implemented in several ways.

One option is at the *algorithmic* level (Finn et al., 2017), where different models are learned for different tasks, and an overall algorithm selects the correct model for a particular task. Implementing task conditioning at the algorithmic level is not optimal, since in most cases, there is a lot of shared knowledge between different (sub-)tasks, which is not exploited when separate models are optimized for each task. Another way of implementing task conditioning is at the *architecture* level. For bidirectional STR, Shi et al. (2018) have implemented the decoding direction as two sub-tasks at the architecture level, that is, by having two separate decoders: one for left-to-right and another one for right-to-left text decoding. Although both decoders share the same encoder, two separate decoders are optimized for two tasks that are (almost) identical and share the same output space.

Implementing bidirectional decoding at the architecture level is not uncommon, and has been done for other tasks besides STR (Zhang et al., 2018; Zhou et al., 2019). However, having two separate decoders for two tasks that are similar (i.e., left-to-right and right-to-left STR) is not desirable:

(i) From a computational point of view: The two network components do not share weights, which requires separate optimization for both parts.

(ii) From a multi-task learning point of view: There is a lot of shared knowledge between left-to-right and right-to-left decoding and both tasks share the same output space, which is not utilized when optimizing the two decoders apart from each other.

Therefore, the question remains: *Can we have the benefits of bidirectional decoding (left-to-right and right-to-left decoding) for STR without implementing this at the architecture or algorithmic level?*

There is promising room for improvement on the implementation side of bidirectional decoding for STR by just using one decoder for both decoding directions. Instead of implementing this decoding direction at the algorithmic or at the architecture level, we propose a new way of implementing task conditioning, namely, at the *input* level. Implementing task conditioning at the input level means that extra feature information is added to the input of the model. This context information should be exploited by the model so as to condition on the right (sub-)task.

More specifically, the transformer architecture, as used by Sheng et al. (2018) for the encoder-decoder part for STR, has no recurrent inductive bias. To solve sequential problems with transformers at the input level, additional position

embeddings are added to provide the model with information about the order of the input sequence. Due to the "position unawareness" of the transformer, the model is also not limited to an inductive decoding direction (unlike RNNs). By adding an extra embedding to the input data, which tells the method to decode an input example from left-to-right or right-to-left, the model can exploit bidirectional decoding with one unified architecture for both directions. This means that the model has one decoder with one set of model parameters that can be optimized for both subtasks at the same time. This is in contrast with the method by Shi et al. (2018), where two decoders are optimized, one for each decoding direction.

In this chapter, we show that we can simplify the bidirectional STR architecture by using a transformer-based encoder-decoder which is able to perform bidirectional text recognition by using a single decoder. Our main technical contributions in this chapter are the following:

- We introduce Bi-STET, **BI**directional **S**cene **TE**xt **T**ransformer. Bi-STET is a unified network architecture, optimized for two sub-tasks (left-to-right and right-to-left STR), using one forward pass. We achieve this through the implementation of bidirectional decoding at the input level as opposed to previous works that do this at the architecture level (Shi et al., 2018). We condition the output sequence on a specific decoding direction by adding extra features at the input level, which results in a direction-agnostic decoder architecture. It is possible to exploit the transformer architecture for task conditioning at the input level, without requiring additional model components or algorithms to model this task conditioning.
- We show that Bi-STET achieves or outperforms state-of-the-art STR methods with a simpler and more efficient approach than other bidirectional STR methods. We achieve these similar results with fewer weight parameters and 50% less training iterations.
- We provide analyzes and insights on the performance of Bi-STET.[2] We analyze the generalisation of Bi-STET w.r.t. oriented and curved text, the learned attention mechanism of the encoder-decoder and the relation between sequence length and test-accuracy of Bi-STET.

2 For reproducibility and repeatability, the code and checkpoint files used to train and evaluate Bi-STET will be made available at `https://github.com/MauritsBleeker/Bi-STET`.

3.2 RELATED WORK

Traditional methods for STR (Shivakumara et al., 2011; Bissacco et al., 2013) apply a bottom-up approach. The input image is preprocessed for feature extraction and character segmentation is applied to obtain single characters from the input image for word inference. For an overview, see (Zhu et al., 2016). With the rise of deep learning, STR methods increasingly focus on end-to-end training from the input image to the desired output character sequence.

3.2.1 Deep-learning based text recognition

Jaderberg et al. (2014a) have proposed the first method for unconstrained STR with deep learning. The method predicts a sequence of characters with a fixed length by using a CNN classification model. A bag-of-N-grams is also predicted; representing an unordered set of character N-grams that occur in the word depicted in the input image. The predictions are combined in a path select layer to predict the most likely character sequence. More recently, Jaderberg et al. (2016) propose another method where the recognition task is formulated as a multi-class classification problem over a 90k-class lexicon.

Many other STR methods (Shi et al., 2016; Cheng et al., 2017; Shi et al., 2017; Shi et al., 2018; Yang et al., 2019; Zhan and Lu, 2019) use a CNN for feature extraction in combination with an encoder-decoder model to map the sequence of image features to a character sequence. To solve the problem of rotation invariance of CNNs (i.e., for curved and oriented text), several methods have been proposed (Shi et al., 2016; Yang et al., 2017; Shi et al., 2018; Yang et al., 2019; Zhan and Lu, 2019). Shi et al. (2018), Shi et al. (2016) and Yang et al. (2019) use a spatial transformer network (STN) for input image rectification to handle perspective text and curved text. Zhan and Lu (2019) have introduced an iterative rectification network where the input image is rectified multiple times by removing perspective distortion and text line curvature.

The alignment between the predicted character and the corresponding region in the input image is modeled with two different approaches. The first approach is to use CTC loss (Graves et al., 2009; Su and Lu, 2014; Liu et al., 2016; Gao et al., 2017; Shi et al., 2017). The other is to connect the RNN encoder to the decoder via an attention mechanism (Shi et al., 2016; Cheng et al., 2017; Yang et al., 2017; Shi et al., 2018; Zhan and Lu, 2019), which creates an addi-

tional context vector for the encoded input image conditioned on the already predicted output sequence.

Shi et al. (2018) introduce the notion of bidirectional STR. Each output sequence is predicted in two directions with two separate decoders, which do not share parameters. The output sequence with the highest probability is selected as the final prediction in order to obtain more robust predictions. Sheng et al. (2018) are the first to use a non-recurrent encoder-decoder approach based on a transformer architecture; they have also introduced a modality-transform block to map an image to a sequence feature representation.

In contrast to most work in STR, we do not use any specific component for image rectification. We also do not rely on RNNs for sequence modeling. Similar to Sheng et al. (2018), we also use a transformer architecture which yields state-of-the-art results in text recognition without using extra image rectification components, for bidirectional sequence modelling STR. However, we achieve this by using a single decoder for the bidirectional decoding, resulting in significantly fewer training iterations.

3.2.2 Task conditioning

As indicated by Radford et al. (2019), the distribution over the possible model outputs is naturally modelled as $p(y \mid x)$, where x is the input data, and y is a possible output prediction. With *task conditioning* (or modality conditioning), the output is not only conditioned on the input data, but also on a given task, dataset, or modality as well, i.e., $p(y \mid x, t)$, where t stands for the *task*.

One way of implementing task conditioning is at the *architecture* level. Kaiser et al. (2017) introduce a single model for eight tasks; for each data-modality and/or subtask, different encoders and decoders are used with a shared latent space. Devlin et al. (2019) adopt the transformer to train a task-agnostic language model. After training the language model, additional task-specific output layers are fined-tuned for each evaluation task. Finn et al. (2017) introduce a meta-learning framework for multi-task learning where task conditioning is implemented at the *algorithmic* level: tasks are treated as training examples and are sampled from a distribution over tasks during training. During each training iteration, for each task, a separate model for each task is updated based on the loss for that specific task.

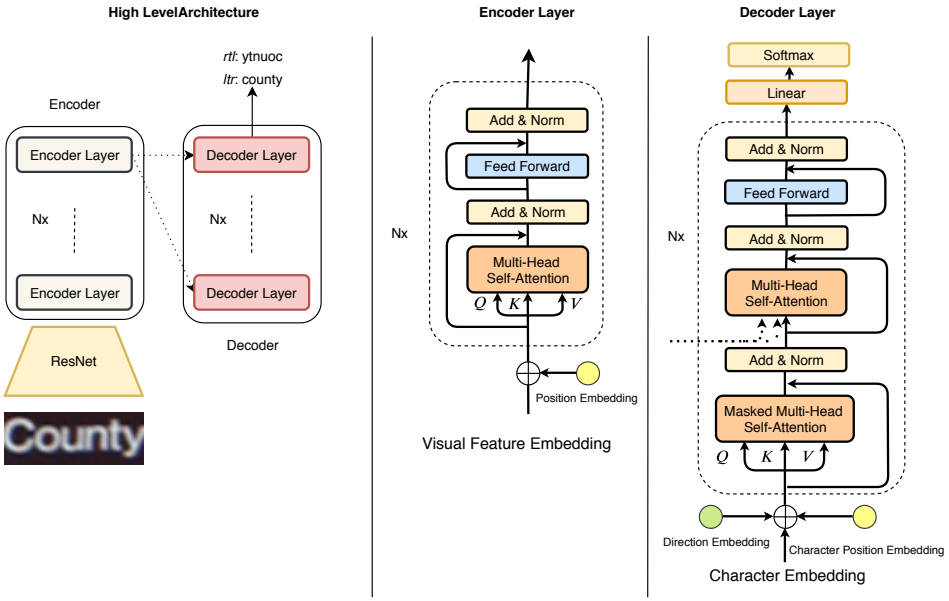


Figure 3.1: Overview of Bi-STET. A ResNet architecture is used for visual feature extraction. Next, a stack of n transformer encoder layers is used for encoding the visual image features. For decoding the output sequence, a stack of n decoder transformer layers is used.

Unlike previous work, we condition a subtask (i.e., the decoding direction) at the input level. As a result, we do not need different models or network components for each decoding direction. Having only one decoder is desirable from both a computational point of view (i.e., only one decoder to optimize) and from an optimization point of view (i.e., shared weights for all sub-tasks).

3.3 METHOD

To address the STR task, we take a fixed size image $\mathcal{I}$ as input and want to decode the sequence of output characters $y_1, \ldots, y_L$, where L is the length of the character sequence depicted in the input image. Briefly, we use a multi-layer stack of transformers for both the encoder and decoder. We use exactly the same implementation of the transformer as described in (Vaswani et al., 2017) for the encoder-decoder. Therefore, we refer to (Vaswani et al., 2017) for the exact details of the implementation. A full overview of Bi-STET is shown in Figure 3.1.

3.3.1 Visual feature extraction network

Like (Cheng et al., 2017; Shi et al., 2018; Yang et al., 2019; Zhan and Lu, 2019), we use a ResNet-based (He et al., 2016) architecture for the visual feature extraction network (VFEN). A ResNet architecture is a more suitable feature extractor than VGG (Simonyan and Zisserman, 2015) for STR, as shown in (Shi et al., 2018; Zhan and Lu, 2019). We use a 45-layer residual network, with the same network configuration as (Shi et al., 2018). We split the obtained feature representation $\mathcal{Q} \in \mathbb{R}^{W \times C \times H}$ column-wise, which results in a sequence of visual feature embeddings (VFEs), $\mathbf{v}_1, \ldots, \mathbf{v}_W$, where $\mathbf{v}_i \in \mathbb{R}^{C \times H}$.

3.3.2 Feature encoding

The feature encoder is in charge of encoding the visual image embeddings. Each visual image embedding is encoded into a new representation in n steps, by using transformer encoder layers, while attending over the entire sequence of VFEs during each encoding step. We use scaled dot-product as the attention function:

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}. \tag{3.1}$$

Scaled dot-product attention can be described as a weighted sum of the vectors in matrix $\mathbf{V}$, which is a horizontal concatenation of the flattened sequence of VFEs (also referred to as *values*). Each embedding $\mathbf{v}$ is weighted by the similarity between a key $\mathbf{k}$ and a query $\mathbf{q}$. In each transformer layer multiple heads of attention are used:

$$head_i = \text{attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \tag{3.2}$$

and

$$MultiHeadSelfAttention = ConCat(head_1, \ldots, head_h)\mathbf{W}^O, \tag{3.3}$$

An advantage of using multiple attention heads is that it allows the model to learn to attend over different positions in the input image per attention head during each step of the encoding process. For self-attention during encoding, the matrices $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are consistent per layer (i.e., $\mathbf{Q} = \mathbf{K} = \mathbf{V}$) and obtained from the output of the previous layer. In the first layer, they can be obtained from the VFEN. The weights of each transformer layer are not shared between

encoder layers. We apply the positional encoding as introduced in (Vaswani et al., 2017).

3.3.3 Character decoding

The decoder consists of n transformer decoder layers. For both the encoder and the decoder we use a transformer architecture. The reason to choose a transformer over an RNN-based architecture is that an RNN already has an inductive bias in terms of decoding and encoding direction due to the recurrent nature of the architecture. The decoder takes the embeddings of the decoded output character sequence as input. Each decoder layer consists of three sublayers: two multi-head attention layers and one feed-forward neural network (the same implementation as in Section 3.3.2). The first multi-head attention layer attends over the decoded output characters (*decoder self-attention*). The second layer of multi-head attention (*decoder cross-attention*) attends over the encoded VFEs from the last encoder layer. The decoder cross-attention is able to look at the encoded input image at every step during decoding. This makes it possible to attend over different encoded image regions during decoding. Previous work on STR (Cheng et al., 2017; Shi et al., 2018; Zhan and Lu, 2019) only uses one attention distribution over the encoded states per decoding step. In contrast, per decoding layer n, we have h attention heads modeling complex alignments between encoder features and decoded output characters.

We add an extra direction embedding in order to add more context information by using additional embeddings. This direction embedding tells the model to decode the output sequence from left-to-right or from right-to-left. By adding the direction embedding, we can use the same decoder network and still condition on the output sequence reading direction.

For every decoding step t, the output embedding $\mathbf{h}_n$ of the stack of transformer decoders is passed through a feed-forward layer with the output characters as the output space. A softmax is applied to obtain a distribution over all output characters. During training, this results in a $V \times L$ matrix, where V is the size of the output character space (or vocabulary) and L the length of the predicted character sequence.

3.3.4 Direction embedding

We define the decoding direction of the output sequence as two sub-tasks of STR. Each decoding direction is one sub-task of the method on which we condition the output sequence. To condition the output on a decoding direction, we randomly initialize two 512-d vectors at the start of training. During each training iteration, every input image in the batch is decoded twice; once from left-to-right and right-to-left. The ground truth description of the right-to-left decoded character sequence is just the reserved ground truth of the original description. During decoding, we add the direction embedding on top of the positional embedding and the token embedding. This is another way to provide additional context information to the model, similar to the position embeddings. Based on this information, the model should learn to decode the character sequence not only in the left-to-right direction but also in the other direction, otherwise, the loss function for the right-to-left decoded images will not be minimized. Similar to the character embeddings, the direction embeddings are trained end-to-end with the rest of the model.

3.4 EXPERIMENTAL SETUP

3.4.1 Datasets

Bi-STET is trained on two synthetically generated datasets. After training, the method is evaluated on seven real-world evaluation sets which are commonly used for scene text recognition.

Training datasets

- *Synth90K.* The *Synth90K dataset* (Jaderberg et al., 2014b) is a synthetically generated dataset for text recognition. It contains 7.2 million training images. The lexicon used contains 90,000 words. Each word has been used to render 100 different synthetic images.
- *SynthText.* The *SynthText dataset* (Gupta et al., 2016) contains 800,000 synthetically generated images for text detection and recognition, with roughly 8 million annotated text instances placed in natural scenes. We crop all text instances from the original input images, by taking the smallest horizontally

aligned bounding box around the annotated text instances in the image. We discard bounding boxes that are smaller than 32 pixels in height or 30 pixels in width. Bounding boxes larger than 800 pixels in width, 500 pixels in height or with a transcription label longer than 25 characters are removed too. We obtain 2.9 million cropped-word images from this dataset for training.

Evaluation datasets

Bi-STET is evaluated cross-dataset. The model is trained only on synthetically generated word images while we evaluate on real-world word images. Unless stated otherwise, we use the word-image crops and annotations as provided by the dataset to be consistent with other methods. This might result in over-cropped word images; in other cases adding a margin may lead to other artifacts.

- *ICDAR03.* The *ICDAR03 dataset* (Lucas et al., 2003) contains 258 images for training and 251 for testing. For the text recognition task only, 1,156 word instances can be cropped from the test set. This dataset was collected for the text detection and recognition task. Therefore, most text instances in the images are clearly horizontally visible and centred in the image (Veit et al., 2016). Following (Wang and Belongie, 2010; Wang et al., 2011; Cheng et al., 2017; Shi et al., 2018), we ignore all words that are shorter than three characters or contain non-alphanumeric characters during evaluation.
- *ICDAR13.* The *ICDAR13 dataset* (Karatzas et al., 2013) contains most images from the ICDAR03 dataset. In total, this dataset contains 1,095 word images for evaluation. Similar to (Shi et al., 2018), we add a cropping margin of 15% to prevent over-cropping.
- *ICDAR15.* The *ICDAR15 dataset* (Karatzas et al., 2015) contains 2,077 word images for evaluating. The word images are cropped from video frames collected with the Google Glass device. These frame crops contain substantial real-world interference factors such as occlusions, motion blur, noise, and illumination factors, which are not present in the ICDAR03 and ICDAR13 datasets. Similar to Cheng et al. (2017), we remove all examples where the ground truth transcription contains non-alpha numeric characters.
- *SVT.* The *Street View Text dataset* (SVT) (Wang and Belongie, 2010; Wang et al., 2011) contains images that have been taken from Google Street View. Due to this origin, some images have a low resolution and/or contain distortion

factors such as noise or blur. This dataset contains 647 word images for evaluation. Per image, a 50-word lexicon is provided as well. Similar to (Shi et al., 2018), we add a cropping-margin of 5% to prevent over-cropping. Similar to (Shi et al., 2018), we add a cropping-margin of 5% to prevent over-cropping.

- *SVTP.* The *Street View Text Perspective dataset* (SVTP) (Phan et al., 2013) contains 645 word images cropped from Street View. Most images have perspective distortions due to the camera viewpoint angle.
- *IIIT-5K Word.* The *IIIT-5k Word dataset* (IIIT5K) (Mishra et al., 2012) contains 3,000 images for evaluation. The word images are cropped from scene texts and born-digital images. For this dataset, per evaluation image, two lexicons of 50 and 1,000 words are provided for lexicon inference.
- *CUTE80.* The *Curved Text dataset* (CUTE80) (Risnumawan et al., 2014) mainly contains curved and/or oriented text instances. The dataset was originally proposed for text detection but later annotated for text recognition as well. In total, 288 high-resolution word images can be cropped from the original dataset.

3.4.2 Implementation details

Our implementation consists of a feature extraction network followed by an encoder-decoder network. The code and checkpoint files used to train and evaluate Bi-STET are available at `https://github.com/MauritsBleeker/Bi-STET`.

Feature extraction network

All input images are resized to 32×256 without keeping the original aspect ratio. The maximum output sequence length during training is 24. All pixels are normalized with a per-channel calculated mean and standard deviation calculated on the ImageNet dataset (Deng et al., 2009).

Encoder-decoder network

For the encoder and decoder, we use exactly the same configuration as the base model described in (Vaswani et al., 2017). We use a stack of $n = 6$ transformer layers for both the encoder and decoder. Each layer has eight attentions heads ($h = 8$). The embedding dimensionality is set to $d = 512$. For the hidden

state of the two layer feed-forward network in each transformer layer, we set $d_f = 2048$.

The output space of our model contains all the lower-case characters $\{a, \ldots, z\}$, digits $\{0, \ldots, 9\}$, 32 ASCII punctuation marks, similar to (Cheng et al., 2017; Shi et al., 2018; Zhan and Lu, 2019), and a start- and end-of-word symbol. The punctuation marks are included during training, but ignored during evaluation. All evaluation and training ground truth descriptions are lower-case, which makes the model case-insensitive.

3.4.3 Optimization

The entire method is trained from scratch. All the weights are initialized with Xavier initialization (Glorot and Bengio, 2010). Similar to (Shi et al., 2016; Cheng et al., 2017; Shi et al., 2017; Yang et al., 2017; Shi et al., 2018; Zhan and Lu, 2019), we use ADADELTA (Zeiler, 2012) as the optimizer for the model. ADADELTA has a self-adaptable learning rate, which we initialize to 1 Even though the learning rate of ADADELTA is self-adaptable, we apply a learning rate schedule where we reduce the initial learning rate by a factor of 0.1 after 150,000, 300,000 and 400,000 training iterations. Similar to (Shi et al., 2018), we find that a learning rate schedule is beneficial to the performance.

The model is trained for 500,000 training iterations in total, after which it converges. We use Kullback-Leibler divergence as the loss function. The batch size is set to 64. For each training batch, we sample 32 images from the Synth90k dataset and 32 from the SynthText. Shi et al. (2018) and Zhan and Lu (2019) show that methods optimized with balanced batch (of size 64) on the SynthText and Synth90k datasets outperform methods solely trained on Synth90k. Per forward-backward pass, we decode the characters per example from left-to-right and from right-to-left.

During training, we do one forward pass for left-to-right decoding and one for right-to-left and accumulate the gradients. It is possible to train both decoding directions with one forward pass, but for computational reasons, we have chosen gradient accumulation instead.

3.4.4 Metrics

We use the same evaluation metrics as in (Cheng et al., 2017; Shi et al., 2018; Zhan and Lu, 2019). The text recognition task includes 68 characters in total. During evaluation, the 32 ASCII punctuation marks are ignored. When a lexicon is provided, the word from the lexicon with the shortest edit distance is selected as the prediction. Only predicted sequences of characters that are completely correct are considered to be correctly predicted examples. We select the character with the highest probability per index in the sequence, until the end-of-word character is predicted. When decoding bidirectionally, the sequence with the highest product probability is selected as the final output sequence.

3.5 RESULTS

First, we compare bidirectional sequence predicting for STR with a single decoder vs. with two decoders. Next, we examine the performance of Bi-STET and other models on STR evaluation sets. Finally, we provide analyzes of the attention mechanism in Bi-STET and the capability of the method to handle curved and rotated text.

3.5.1 Bidirectional decoding

Similar to (Shi et al., 2018), we condition the output character sequence on a decoding direction. We validate our universal bidirectional decoding with three evaluation variants, similar to Shi et al. (2018). For the first variant, we decode the output sequence from left-to-right, by only using the left-to-right direction embedding. In the second variant, we only use the right-to-left directional embedding. In the third variant, we decode each evaluation example twice, once with each direction embedding. The two predicted outputs can have different sequence lengths. For each prediction (left-to-right and right-to-left) we take the sequence with the highest probability by taking the arg-max for each position and take the product of the probabilities as the probability of the entire sequence. We select the sequence with the highest output probability as the final prediction. In case that the right-to-left prediction has the highest probability, we reverse the sequence to match with the ground truth.

Table 3.1: Accuracy of left-to-right vs. right-to-left vs. bidirectional word decoding. Measured without lexicon and compared with the method by Shi et al. (2018).

Method	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE
Shi et al. (2018), left-to-right	91.93	88.76	93.49	89.75	–	74.11	73.26
Shi et al. (2018), right-to-left	91.43	89.96	92.79	89.95	–	73.95	74.31
Shi et al. (2018), bidirectional	92.27	**89.5**	93.60	90.54	–	74.26	74.31
Bi-STET (this chapter), left-to-right	94.2	88.3	95.1	92.5	75.0	78.8	81.8
Bi-STET (this chapter), right-to-left	94.1	87.9	95.3	**93.4**	73.2	79.5	**83.6**
Bi-STET (this chapter), bidirectional	**94.7**	89.0	**96.0**	**93.4**	**75.7**	**80.6**	82.5

In Table 3.1 we show the results of the three aforementioned evaluation variants and the results obtained by Shi et al. (2018). For 6 out of 7 evaluation sets, we achieve state-of-the-art results for bidirectional STR. For 6 out of 7 of the evaluation sets, Bi-STETs bidirectional decoding leads to higher scoring sequence prediction than using a single decoding direction. Only for the CUTE80 set, the right-to-left decoding leads to a higher accuracy score than bidirectional. The gain in performance due to the bidirectional decoding is similar as in the method by Shi et al. (2018).

We also show that, by using a transformer-based encoder-decoder, bidirectional STR can be substantially simplified in comparison to the method by Shi et al. (2018). In Table 3.2, we compare the number of model parameters and the number of training iterations with the method by Shi et al. (2018). We use similar training settings, in terms of batch size, optimization, data, etc. as (Shi et al., 2018). Based on Table 3.2, it is clear that with a single bidirectional transformer decoder, the number of training iterations can be reduced by 50% compared to methods that use two separate RNN decoders – in combination with significantly fewer model parameters.[3] Fewer training iterations and model parameters are excellent properties from an efficiency and computational point of view. We also outperform the RNN-based method, which also uses an extra image rectification network, on most evaluation sets.

3 In hindsight, it turned out that we overestimated the parameter count of the method by Shi et al. (2018) with a factor of three (the provided model checkpoint also included parameter gradients and activations). Therefore, the claims we make w.r.t. parameter efficientcy are incorrect, and different from those in (Bleeker and de Rijke, 2020).

Table 3.2: Comparison of the number of trainable model parameters and training iterations.

Method	Model parameters ($\times 10^6$)	Training iterations ($\times 10^6$)	Batch size
Shi et al. (2018)	88	1	64
Bi-STET (this chapter)	**66**	**0.5**	64

To summarize, the results in Table 3.1 and Table 3.2 show that similar or better results can be obtained with significant less training parameters and twice the efficiency by using a single transformer decoder.

3.5.2 Text recognition

In Table 3.3, we evaluate Bi-STET in terms of prediction accuracy on 7 public evaluation sets and compare it to other state-of-the-art (SOTA) STR methods. Bi-STET meets or outperforms SOTA methods on 6 out of 12 evaluation experiments. We achieve new SOTA results on the ICDAR03 and the IIIT5K datasets.

The strength of the transformer encoder-decoder w.r.t. to images with oriented and curved text is also shown by the results in Table 3.3. The five datasets where Bi-STET does not beat, but meets, the state-of-the-art are CUTE80, ICDAR13, ICDAR15, SVT-P and SVT. The fact that we do not achieve state-of-the-art on the datasets SVT-P, ICDAR15 and CUTE80 can be explained by the fact that those datasets contain a considerable number of images that are rotated or have perspective distortions. We meet SOTA results on these datasets, we speculate that we do not exceed them because they have these distortions. It should also be noted we are able to meet these SOTA results without any specific network component for dealing with distortions, which other methods explicitly require (Shi et al., 2018; Yang et al., 2019; Zhan and Lu, 2019). For ICDAR13 and SVT-P we do not establish new SOTA performance figures, although we do meet the results of other methods with a small margin.

Table 3.3: Accuracy compared to state-of-the-art. ST is short for the SynthText dataset, 90K for the Synth90K dataset; 50, 1k, full and 0 are the size of the used lexicons; 0 means that no lexicon is used.

Method	ConvNet, Data	IIIT5k			SVT		IC03			IC13	IC15	SVTP	CUTE
		50	1k	0	50	0	50	Full	0	0	0	0	0
Su and Lu (2014)	–	–	–	–	83.0	–	92.0	82.0	–	–	–	–	–
Jaderberg et al. (2016)	VGG, 90k	97.1	92.7	–	95.4	80.7	98.7	98.6	93.1	90.8	–	–	–
Jaderberg et al. (2014a)	VGG, 90k	95.5	89.6	–	93.2	71.7	97.8	97.0	89.6	81.8	–	–	–
Shi et al. (2017)	VGG, 90k	97.8	95.0	81.2	97.5	82.7	98.7	98.0	91.9	89.6	–	–	–
Shi et al. (2016)	VGG, 90k	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6	–	71.8	59.2
Lee and Osindero (2016)	VGG, 90k	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0	–	–	–
Yang et al. (2017)	VGG, Private	97.8	96.1	–	95.2	–	97.7	–	–	–	–	75.8	69.3
Cheng et al. (2017)	ResNet, 90k+ST$^+$	99.3	97.5	87.4	97.1	85.9	**99.2**	97.3	94.2	93.3	70.6	–	–
Shi et al. (2018)	ResNet, 90k+ST	**99.6**	98.8	93.4	97.4	89.5	98.8	98.0	94.5	91.8	76.1	78.5	79.5
Zhan and Lu (2019)	ResNet, 90k + ST	**99.6**	98.8	93.3	97.4	**90.2**	-	-	-	91.3	76.9	79.6	83.3
Sheng et al. (2018)	Modality-Transform, 90k	99.2	98.8	86.5	**98.0**	88.3	98.9	97.9	95.4	**94.7**	-	-	-
Yang et al. (2019)	ResNet, 90k+ST	99.5	98.8	94.4	97.2	88.9	99.0	98.3	95.0	93.9	**78.7**	**80.8**	**87.5**
Bi-STET (this chapter)	ResNet, 90k+ST	**99.6**	**98.9**	**94.7**	97.4	89.0	99.1	**98.7**	**96.0**	93.4	75.7	80.6	82.5

3.5.3 Analyzes

Attention head analyzes

To get an understanding of the internal behaviour of Bi-STET, we extracted the attention distributions from Bi-STET during evaluation and visualize them in Figures 3.2a and 3.2b. Conditioned on different decoding directions, Bi-STET has learned to model an inverse alignment between the predicted output character and the region in the input images where the character is depicted – using only a single decoder. In Figure 3.2a, there is a clear attention alignment going from left to right over the image, while in Figure 3.2b, this alignment goes in the opposite direction. The model has jointly learned to model the character-image region alignment in both directions. Also, different attention heads do not specialize for left-to-right or right-to-left decoding, but learn how to change the attention direction when the output is conditioned on a different decoding direction. This shows the strength of the method w.r.t. regularisation towards both sub-tasks. This is interesting from a multi-task learning point of view because this indicates that the same attention heads can learn different (sub)-tasks.

Rotated and curved text

Bi-STET is solely trained as a general image-to-text encoder-decoder and does not contain a specific rectification component for handling rotated or curved

(a) Decoding attention while decoding from left-to-right.

(b) Decoding attention while decoding from right-to-left.

Figure 3.2: Visualization of the self-attention of layer 5 of the decoder while decoding the sequence (left-to-right). Each row in the matrix visualizes the attention distribution over the embeddings of the image on the X-axis, while decoding the corresponding character in the input.

text instances, unlike previous methods (Shi et al., 2018; Yang et al., 2019; Zhan and Lu, 2019). We are able to obtain results that meet those of state-of-the-art methods that are specifically optimized for curved and rotated text. Figure 3.3 provides a sample from the CUTE80 dataset with correctly and incorrectly predicted sequences. Looking at correctly predicted examples, we see that Bi-STET properly decodes words that are slightly curved or only curved in one direction. This is where the bidirectional decoding shows its strength. For example, the two middle images of the second row are correctly decoded when decoding from right-to-left, but not when decoding in the other direction. From the first row of images, we see that words that are curved in very strong arc shapes (heavy perspective distortions) are difficult for Bi-STET to decode. This also shows the strength of method w.r.t. regularisation towards curved and rotated text without using any specific rectification component.

Sequence length

Vaswani et al. (2017) argue that transformer-based architectures are more suitable for capturing long-range dependencies for machine translation than RNNs, because of the global attention per encoding and decoding step. The self-attention results in the fact that the maximum distance in a sequence between two embeddings which are encoded or decoded is 1. Despite the fact that the maximum output sequence length in our evaluation experiment (max. length 17) is not as long as for other language tasks (Khandelwal et al., 2018),

(a) ballys vs. bally **(b)** chathamkent vs. from **(c)** starbucks vs. and **(d)** coffee vs. offer

(e) ronaldo vs. ronaldo **(f)** meant vs. meant **(g)** football vs. football **(h)** united vs. united

Figure 3.3: Examples of curved text examples from the CUTE80 dataset that are correctly and incorrectly predicted by Bi-STET. In black the ground truth is given.

we are interested in whether or not our transformer-based method is better at predicting longer output character sequence than an RNN-based method. In Figure 3.4, we show the relation between output sequence length and accuracy for the IIIT-5K evaluation set. We can see that the prediction accuracy given a sequence length is more or less constant until we reach a character length of 11. After a sequence length of 11, the accuracy starts to degrade. It should be noted that there are very few samples with a sequence length of 11 or higher.

By comparing Figure 3.4 to Figure 12 in (Shi et al., 2018), we see that Bi-STET performs similarly to Shi et al. (2018)'s method for short character sequences and slightly better for longer character sequences which are longer than 11 characters. We conclude that scene text transformer (STET) performs similarly for short character sequences and at least as good in decoding longer character sequences.

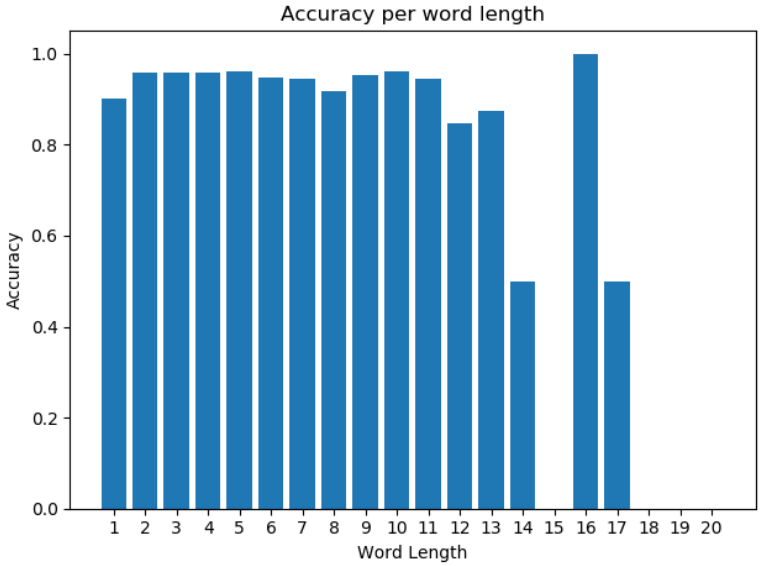


Figure 3.4: Text recognition accuracies versus word length for Bi-STET. Tested on IIIT-5K.

3.6 DISCUSSION AND CONCLUSION

In this chapter, we answered the second research question of this thesis positively by introducing Bi-STET: a method that unifies bidirectional STR with a single decoder architecture. Bi-STET is capable of bidirectional decoding, without implementing the decoding direction conditioning at the architecture or algorithmic level. The decoding conditioning is implemented at the input level, by adding an extra direction embedding to the input.

We show that Bi-STET achieves or outperforms state-of-the-art STR methods, with a considerably more efficient approach than other bidirectional STR methods (i.e., requiring 50 % less training iterations and significant less model parameters.[4] By having fewer model parameters, the model can be executed on devices with less computational resources (for user applications). Besides that, less computational resources are required to obtain SOTA text recognition results. We also show that Bi-STET learns to exploit the same attention heads for both decoding directions, which means that there are no specialized attention heads in the model for each decoding direction. This is interesting from a multi-task learning point of view because different heads tend not to be focused on one decoding direction. Finally, we show that, due to the bidirec-

4 See footnote 3.

tional decoding, Bi-STET is capable of handling slightly curved and orientated text and performs as well for longer text sequences as other bidirectional STR methods.

A future research direction is to combine Bi-STET with a spatial transformer network (Shi et al., 2016; Shi et al., 2018; Yang et al., 2019) or a rectification network (Zhan and Lu, 2019). Bi-STET is able to handle oriented and perspective text in images; we believe that Bi-STET could benefit from an extra image processing component to be able to better handle oriented or perspective text. From a multi-task learning point of view, it would be interesting to explore task conditioning on the input level with more diverse tasks. In addition, an extension to tasks with more complex and diverse data modalities would also be a possible future research direction.

This chapter concludes Part 1 of this thesis, which has a focus on multi-modal sequence modeling. In Part 2, we continue with multi-modal representation learning for image-text matching. Specifically, we focus on contrastive representation methods and the image-caption retrieval evaluation task.

Part 2 - Image-Text Representation Learning

4

Do Lessons from Metric Learning Generalize to Image-Caption Retrieval?

In the second part of this thesis, our attention shifts away from multi-modal sequence modeling and we focus on multi-modal representation learning instead. Specifically, we focus on representation learning for images and text. In image-text representation learning, the goal is to learn general-purpose data representations of the images and text that generalize well to various downstream evaluation tasks. Throughout the second part of this thesis, the primary multi-modal task we use to evaluate the image and text representations is the image-caption retrieval (ICR) task. In ICR, an image or caption is used as a query, and the goal is to rank a set of candidates in the other modality.

The high-level setup for image-text representation learning consists of two encoders, one for each modality, that map the images and text into a shared latent space. In metric learning, the goal is to project the input data to a latent space where similar pairs of information are approximately close (in terms of a distance metric) (Musgrave et al., 2020). To learn the embeddings in this semantic latent space, contrastive losses (Hadsell et al., 2006) are a prominent choice of optimization objective. Despite the developments and successes in the metric learning field, few of the proposed loss functions have been tried in the context of ICR. The triplet loss with semi-hard negatives is the de facto choice for optimization function for many ICR tasks. Therefore, in this chapter[1] we raise the following research question:

Research Question 3: *Do lessons from metric learning generalize to image-caption retrieval?*

1 This chapter is based on (Bleeker and de Rijke, 2022).

To answer this research question, we first evaluate three prominent contrastive loss functions in a fair manner on the ICR task: (i) the triplet loss (Kiros et al., 2014), including semi-hard negative mining (Faghri et al., 2018), (ii) the NT-Xent loss (Chen et al., 2020c), and (iii) SmoothAP (Brown et al., 2020a). Surprisingly, we find that the triplet loss outperforms the other two losses. To explain why certain contrastive losses perform better than others, we introduce *counting contributing samples (COCOS)*, a method that counts how many samples contribute to the gradient w.r.t. the query representation. The COCOS method shows that the underperforming contrastive losses take too many (non-informative) negative samples into account for the gradient.

4.1 INTRODUCTION

Given a query item in one modality, *cross-modal retrieval* is the task of retrieving similar items in another modality (Zeng et al., 2020). We focus on *image-caption retrieval* (ICR) (Lee et al., 2018; Li et al., 2019a; Verma et al., 2020; Diao et al., 2021). For the ICR task, given an image or a caption as a query, systems have to retrieve the positive (e.g., matching or similar) item(s) in the other modality. Most ICR methods work with a separate encoder for each modality to map the input data to a representation in a shared latent space (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Diao et al., 2021; Jia et al., 2021). The encoders are optimized by using a contrastive loss criterion, so as to enforce a high degree of similarity between representations of matching items in the latent space. For retrieval, a similarity score between a query and each candidate in a candidate set is computed to produce a ranking with the top-k best matching items. A lot of recent work on ICR relies on:

(i) pre-training on large amounts of data (Li et al., 2020c; Jia et al., 2021; Radford et al., 2021), and

(ii) more sophisticated (and data-hungry) model architectures (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Messina et al., 2020b; Diao et al., 2021).

However, pre-training on large-scale datasets is not always an option, either due to a lack of compute power, a lack of data, or both. Hence, it is important to continue to develop effective ICR methods that only rely on a modest amount of data.

To learn the similarity between a query and candidate representations, most ICR work relies on the standard triplet loss with semi-hard negatives (triplet SH) (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Chen et al., 2020a; Messina et al., 2020b; Diao et al., 2021) or on the cross-entropy based NT-Xent (Chen et al., 2020b; Jia et al., 2021) loss. In *metric learning*, the focus is on loss functions that result in more accurate item representations (in terms of a given evaluation metric) that can distinguish between similar and dissimilar items in a low-dimensional latent space (Musgrave et al., 2020). There has been important progress in metric learning, with the introduction of new loss functions that result in better evaluation scores on a specific (evaluation) task. Examples include SmoothAP (Brown et al., 2020a), a smooth approximation of the discrete evaluation metric average precision. By using SmoothAP, a retrieval method can be optimized with a discrete ranking evaluation metric and can handle multiple positive candidates simultaneously, which is not possible for the standard triplet loss. Loss functions such as SmoothAP narrow the gap between the training setting and a discrete evaluation objective and thereby improve evaluation scores.

Research goal. Most metric learning functions work with general representations of similar/dissimilar candidates and, in principle, there is no clear argument why obtained results on a specific task/method should not generalize to other tasks or methods. Hence we ask the following research question: *can newly introduced metric learning approaches, that is, alternative loss functions, be used to increase the performance of ICR methods*? We compare three loss functions for the ICR task:

(i) the triplet loss (Kiros et al., 2014), including semi-hard negative mining (Faghri et al., 2018),
(ii) NT-Xent loss (Chen et al., 2020c), and
(iii) SmoothAP (Brown et al., 2020a).

We expect SmoothAP to result in the highest performance based on the findings in the context of image retrieval (Brown et al., 2020a) and in representation learning (Varamesh et al., 2020).

Main findings. Following (Musgrave et al., 2020), we evaluate the three loss functions on fixed methods, with different datasets, and with a fixed training regime (i.e., training hyper-parameters) to verify which loss function uses the given training data as effectively as possible. Surprisingly, the lessons from metric learning do not generalize to ICR. The triplet loss with semi-hard neg-

ative mining still outperforms the other loss functions that we consider. The promising results obtained by SmoothAP and the NT-Xent loss in other fields do not generalize to the ICR task.

To get a better grasp of this unexpected outcome, we propose *counting contributing samples (COCOS)*, a method for analyzing contrastive loss functions. The gradient w.r.t. the query for the triplet loss, NT-Xent and SmoothAP can be formulated as a sum over the representations of the positive and negative candidates in the training batch. The main difference between the loss functions lies in the number of samples used when computing the gradient w.r.t. the query and how each sample is weighted. Using this gradient analysis we compare loss functions by counting how many samples contribute to the gradient w.r.t. the query representation at their convergence points. This yields an explanation of why one loss function outperforms another on the ICR task.

Main contributions. In this chapter, we contribute the following:

(i) We experimentally compare three loss functions from the metric learning domain to determine if promising results from metric learning generalize to the ICR task, and find that the triplet loss with semi-hard (SH) negative mining still results in the highest evaluation scores.

(ii) We propose COCOS, a way of analyzing contrastive loss functions, by defining a count that tells us how many candidates in the batch contribute to the gradient w.r.t. the query. On average, the best performing loss function takes at most one (semi-hard) negative sample into account when computing the gradient.

4.2 BACKGROUND AND RELATED WORK

4.2.1 Notation

We follow the notation introduced in (Brown et al., 2020a; Chen et al., 2020c; Varamesh et al., 2020). We start with a multi-modal image-caption dataset $\mathcal{D} = \{(\mathbf{x}_I^i, \mathbf{x}_{C_1}^i, \ldots, \mathbf{x}_{C_k}^i)^i, \ldots\}_{i=1}^N$ that contains N image-caption tuples. For each image $\mathbf{x}_I^i$, we have k matching/corresponding captions, $\mathbf{x}_{C_1}^i, \ldots, \mathbf{x}_{C_k}^i$.

In the ICR task, either an image or a caption can function as a query. Given a query $\mathbf{q}$, the task is to rank all candidates in a candidate set $\Omega = \{\mathbf{v}_i \mid i = 0, \ldots, m\}$. A matching candidate is denote as $\mathbf{v}^+$ and a negative candidate(s)

as $\mathbf{v}^-$. For each query $\mathbf{q}$, we can split the candidate set Ω into two disjoint subsets: $\mathbf{v}^+ \in \mathcal{P}_\mathbf{q}$ (*positive* candidate set) and $\mathbf{v}^- \in \mathcal{N}_\mathbf{q}$ (*negative* candidate set), where $\mathcal{N}_\mathbf{q} = \{\mathbf{v}^- \mid \mathbf{v}^- \in \Omega, \mathbf{v}^- \notin \mathcal{P}_\mathbf{q}\}$. We assume a binary match between images and captions, they either match or they do not match.

The set with similarity scores for each $\mathbf{v}_i \in \Omega$ w.r.t. query $\mathbf{q}$ is defined as: $\mathcal{S}_\Omega^\mathbf{q} = \{s_i = \langle \frac{\mathbf{q}}{\|\mathbf{q}\|} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \rangle, i = 0, \ldots, m\}$. We use cosine similarity as a similarity scoring function. $S_\Omega^\mathbf{q}$ consists of two disjoint subsets: $S_\mathcal{P}^\mathbf{q}$ and $S_\mathcal{N}^\mathbf{q}$. $S_\mathcal{P}^\mathbf{q}$ contains the similarity scores for the positive candidates and $S_\mathcal{N}^\mathbf{q}$ the similarity scores for the negative candidates. During training, we randomly sample a batch $\mathcal{B}$ with image-caption pairs. Both the images and captions will function as queries and candidates.

4.2.2 Image-caption retrieval

The ICR task can be divided into *image-to-text* (i2t) and *text-to-image* (t2i) retrieval. We target specific ICR methods that are optimized for the ICR-task only and satisfy three criteria: (i) The methods we use have solely been trained and evaluated on the same benchmark dataset; (ii) the ICR methods we use compute one global representation for both the image and caption; and (iii) the methods do not require additional supervision signals besides the contrastive loss for optimization. Below we evaluate two ICR methods with different loss functions: VSE++ (Faghri et al., 2018) and VSRN (Li et al., 2019a). In Appendix 4.C we provide a detailed description of VSE++ and VSRN.

VSE++. The best performing method of VSE++ uses a ResNet-152 (He et al., 2016)) to compute a global image representation. The caption encoder is a single directed GRU-based (Cho et al., 2014a) encoder. Faghri et al. (2018) introduce the notion of mining semi hard negative triplets for the ICR task. By using the hardest negative in the batch for each positive pair (i.e. the negative candidate with the highest similarity score w.r.t. the query), their method outperforms state-of-the-art methods that do not apply this semi-hard negative mining.

VSRN. VSRN takes a set of pre-computed image region features as input. A graph convolutional network (Kipf and Welling, 2017) is used to enhance the relationships among the region vectors. The sequence of region feature vectors is put through an RNN network to encode the global image representation. VSRN uses the same caption encoder and loss as (Faghri et al., 2018).

Other methods. Following VSE++ and VSRN, the SGRAF (Diao et al., 2021) and IMRAM (Chen et al., 2020a) methods have been introduced. We do not use these two methods as they either do not outperform VSRN (Chen et al., 2020a) or rely on similar principles as VSRN (Diao et al., 2021). The main recent progress in ICR has been characterized by a shift towards transformer-based (Vaswani et al., 2017) methods. To the best of our knowledge, TREN/TERAN (Messina et al., 2020a; Messina et al., 2020b) and VisualSparta (Lu et al., 2021b) are the only transformer-based ICR methods that are solely optimized using MS-COCO (Lin et al., 2014) or Flickr30k (Young et al., 2014). We do not use transformer-based methods, as optimizing them does not scale well for a reproducibility study with moderately sized datasets. Methods such as OSCAR (Li et al., 2020c), UNITER (Chen et al., 2020d), ViLBERT (Lu et al., 2019) and ViLT-B (Kim et al., 2021) use additional data sources and/or loss functions for training. They focus on a wide variety of tasks such as visual QA, image captioning, and image retrieval.

4.2.3 Loss functions for ICR

In this section, we introduce three loss functions for ICR.

Triplet loss with semi hard negative mining. The triplet loss is commonly used as a loss function for ICR methods (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Messina et al., 2020b; Diao et al., 2021). The *triplet loss with semi-hard negative mining* (triplet loss SH), for a query $\mathbf{q}$ is defined as:

$$\mathcal{L}^{\mathbf{q}}_{TripletSH} = \max(\alpha - s^+ + s^-, 0), \tag{4.1}$$

where α is a margin parameter, $s^- = \max(S^{\mathbf{q}}_{\mathcal{N}})$ and $s^+ = s_0 \in S^{\mathbf{q}}_{\mathcal{P}}$. Here, $\mathcal{S}^{\mathbf{q}}_{\mathcal{P}}$ only contains one element per query. The triplet loss SH over the entire training batch is defined as:

$$\mathcal{L}_{TripletSH} = \sum_{\mathbf{q}\in\mathcal{B}} \mathcal{L}^{\mathbf{q}}_{TripletSH}. \tag{4.2}$$

Triplet loss SH performs a form of soft-negative mining per query by selecting the negative candidate with the highest similarity score w.r.t. the query, we also refer to this as the maximum violating query. For computational efficiency, this soft-negative mining is executed within the context of the training batch $\mathcal{B}$ and not over the entire training set.

As opposed to the definition above, another possibility is to take the triplet loss over all triplets in the batch $\mathcal{B}$. This is the definition of the standard *triplet loss* (Kiros et al., 2014):

$$\mathcal{L}^{\mathbf{q}}_{Triplet} = \sum_{s^- \in S^q_{\mathcal{N}}} \max(\alpha - s^+ + s^-, 0) \tag{4.3a}$$

$$\mathcal{L}_{Triplet} = \sum_{\mathbf{q} \in \mathcal{B}} \mathcal{L}^{\mathbf{q}}_{Triplet}. \tag{4.3b}$$

NT-Xent loss. The *NT-Xent loss* (Chen et al., 2020c) is a loss function commonly used in the field of self-supervised representation learning (van den Oord et al., 2018; Chen et al., 2020c). A similar function has also been proposed by Zhang and Lu (2018) in the context of ICR. The NT-Xent loss is defined as:

$$\mathcal{L}_{NT\text{-}Xent} = -\frac{1}{|\mathcal{B}|} \sum_{\mathbf{q} \in \mathcal{B}} \log \frac{\exp(s^+/\tau)}{\sum_{s_i \in \mathcal{S}^{\mathbf{q}}_{\Omega}} \exp(s_i/\tau)}, \tag{4.4}$$

where τ functions as a temperature parameter. As for the triplet loss formulation: $s^+ = s_0 \in S^{\mathbf{q}}_{\mathcal{P}}$. The major difference between the triplet loss SH is that the NT-Xent loss takes the entire negative candidate set into account.

SmoothAP loss. The average precision metric w.r.t. a query $\mathbf{q}$ and candidate set Ω is defined as:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}^{\mathbf{q}}_{\mathcal{P}}|} \sum_{i \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}} \frac{\mathcal{R}(i, \mathcal{S}^{\mathbf{q}}_{\mathcal{P}})}{\mathcal{R}(i, \mathcal{S}^{\mathbf{q}}_{\Omega})}, \tag{4.5}$$

where $\mathcal{R}(i, \mathcal{S})$ is a function that returns the ranking of candidate $i \in \mathcal{S}$ in the candidate set:

$$\mathcal{R}(i, \mathcal{S}) = 1 + \sum_{j \in \mathcal{S}, i \neq j} \mathbb{1}\{s_i - s_j < 0\}. \tag{4.6}$$

Let us introduce the $M \times M$ matrix D, where $D_{ij} = s_i - s_j$. By using the matrix D, Eq. 4.5 can be written as:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}^{\mathbf{q}}_{\mathcal{P}}|} \sum_{i \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}} \frac{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}, j \neq i} \mathbb{1}\{D_{ij} > 0\}}{1 + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}, j \neq i} \mathbb{1}\{D_{ij} > 0\} + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathbb{1}\{D_{ij} > 0\}}.$$

The indicator function $\mathbb{1}\{\cdot\}$ is non-differentiable. To overcome this problem, the indicator function can be replaced by a sigmoid function:

$$\mathcal{G}(x; \tau) = \frac{1}{1 + e^{\frac{-x}{\tau}}}. \tag{4.7}$$

By replacing the indicator function $\mathbb{1}\{\cdot\}$ by $\mathcal{G}$, the Average Precision metric can be approximated with a smooth function:

$$AP_{\mathbf{q}} \approx \frac{1}{|\mathcal{S}^{q}_{\mathcal{P}}|} \sum_{i \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}} \frac{1 + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}, j \neq i} \mathcal{G}(D_{ij}; \tau)}{1 + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}, j \neq i} \mathcal{G}(D_{ij}; \tau) + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathcal{G}(D_{ij}; \tau)}.$$

This loss function is called *SmoothAP* and has been introduced in the context of image retrieval (Brown et al., 2020a), following similar proposals in document retrieval and learning to rank (Oosterhuis and de Rijke, 2018; Wang et al., 2018; Bruch et al., 2019a; Bruch et al., 2019b). The total loss over a batch $\mathcal{B}$ can then be formulated as follows:

$$\mathcal{L}_{AP} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{q} \in \mathcal{B}} (1 - AP_{\mathbf{q}}). \tag{4.8}$$

In Appendix 4.B we provide an extended explanation of SmoothAP.

4.3 DO FINDINGS FROM METRIC LEARNING EXTEND TO ICR?

In representation learning it was found that the NT-Xent loss outperforms the triplet loss and triplet loss SH (Chen et al., 2020c). For both the image retrieval and representation learning task, results show that SmoothAP outperforms both the triplet loss SH and the NT-Xent loss (Brown et al., 2020a; Varamesh et al., 2020). We examine whether these findings generalize to ICR.

4.3.1 Experimental setup

We focus on two benchmark datasets for the ICR task: the Flickr30k (Flickr30k) (Young et al., 2014) and MS-COCO Captions (MS-COCO) (Lin et al., 2014) datasets. Similar to (Faghri et al., 2018; Li et al., 2019a), we use the split provided by Karpathy and Li (2015) for MS-COCO and the Flickr30k. For details of the specific implementations of VSE++ (Faghri et al., 2018)[2] and (Li et al., 2019a)[3] we refer to the papers and online implementations. Each method is trained for 30 epochs with a batch size of 128. We start with a learning rate of 0.0002 and after 15 epochs we lower the learning rate to 0.00002.

For VSE++, we do not apply additional fine-tuning of the image encoder after 30 epochs. Our main goal is to have a fair comparison across methods, datasets, and loss functions, not to have the highest overall evaluation scores. For VSE++, we use ResNet-50 (He et al., 2016) as image-encoder instead of

2 https://github.com/fartashf/vsepp

3 https://github.com/KunpengLi1994/VSRN

ResNet-152 (He et al., 2016) or VGG (Simonyan and Zisserman, 2015). ResNet-50 is faster to optimize and the performance differences between ResNet-50 and ResNet-152 are relatively small.

The VSRN method comes with an additional caption decoder, to decode the original input caption from the latent image representation, this to add additional supervision to the optimization process. We remove the additional image-captioning module, so as to exclude performance gains on the retrieval tasks due to this extra supervision. In (Li et al., 2019a), the similarity score for a query candidate pair, during evaluation, is based on averaging the predicted similarity scores of (an ensemble of) two trained models. We only take the predicted relevance score of one model. The reason for this is that the evaluation score improvements are marginal when using the scores of two models (instead of one) but optimizing the methods takes twice as long. Therefore, our results are lower than the results published in (Li et al., 2019a). For all the remaining details, we refer to our repository.[4] When optimizing with SmoothAP, we take all the k captions into account when sampling a batch, instead of one positive candidate. For this reason, we have to increase the number of training epochs k times as well to have a fair comparison. For each loss function, we select the best performing hyper-parameter according to its original work.

4.3.2 Experimental outcomes

We evaluate each loss function we described in Section 4.2 given a dataset and method. For ease of reference, we refer to each individual evaluation with an experiment number (#) (see Table 4.1). To reduce the variance in the results we run each experiment five times and report the average score and standard deviation. Similar to (Faghri et al., 2018; Li et al., 2019a), we evaluate using recall@k with $k = \{1, 5, 10\}$, for both the image-to-text (i2t) and text-to-image (t2i) task. We also report the sum of all the recall scores (rsum) and the average recall value. For the i2t task, we also report the mean average precision at 5 (mAP@5) due to the fact we have k positive captions per image query.

4 https://github.com/MauritsBleeker/ecir-2022-reproducibility-bleeker

Table 4.1: Evaluation scores for the Flickr30k and MS-COCO, for the VSE++ and VSRN.

			i2t					t2i				
Loss function	#	hyper param	R@1	R@5	R@10	average recall	mAP@5	R@1	R@5	R@10	average recall	rsum
			Flickr30k									
			VSE++									
Triplet loss	1.1	$\alpha = 0.2$	30.8±.7	62.6±.3	74.1±.8	55.9±.3	0.41±.00	23.4±.3	52.8±.1	65.7±.3	47.3±.1	309.4±0.9
Triplet loss SH	1.2	$\alpha = 0.2$	**42.4**±.5	**71.2**±.7	**80.7**±.7	64.8±.6	0.50±.01	**30.0**±.3	**59.0**±.2	**70.4**±.4	53.1±.2	**353.8**±1.6
NT-Xent	1.3	$\tau = 0.1$	37.5±.6	68.4±.6	77.8±.5	61.2±.3	0.47±.00	27.0±.3	57.3±.3	69.1±.2	51.1±.2	337.1±1.3
SmoothAP	1.4	$\tau = 0.01$	**42.1**±.8	**70.8**±.6	**80.6**±.8	64.5±.4	0.50±.00	29.1±.3	58.1±.1	69.7±.2	52.3±.2	350.4±1.7
			VSRN									
Triplet loss	1.5	$\alpha = 0.2$	56.4±.7	83.6±.6	90.1±.2	76.7±.5	0.63±.01	43.1±.3	74.4±.3	83.1±.4	66.9±.3	430.7±1.8
Triplet loss SH	1.6	$\alpha = 0.2$	**68.3**±1.3	**89.6**±.7	**94.0**±.5	84.0±.5	0.73±.01	**51.2**±.9	**78.0**±.6	**85.6**±.5	71.6±.6	**466.6**±3.3
NT-Xent	1.7	$\tau = 0.1$	50.9 ±.5	78.9±.7	86.6±.4	72.2±.4	0.59±.00	40.6±.6	71.9±.2	81.7±.3	64.7±.2	410.6±1.5
SmoothAP	1.8	$\tau = 0.01$	63.1±1.0	86.6±.8	92.4±.5	80.7±.7	0.69±.00	45.8±.2	73.7±.3	82.3±.2	67.3±.1	444.0±2.1
			MS-COCO									
			VSE++									
Triplet loss	2.1	$\alpha = 0.2$	22.1±.5	48.2±.3	61.7±.3	44.0±.3	0.30±.00	15.4±.1	39.5±.1	53.2±.1	36.0±.1	240.0±0.9
Triplet loss SH	2.2	$\alpha = 0.2$	**32.5**±.2	**61.6**±.3	**73.8**±.3	56.0±.2	0.41±.00	**21.3**±.1	**48.1**±.1	**61.5**±.0	43.6±.1	**298.8**±0.8
NT-Xent	2.3	$\tau = 0.1$	25.8±.5	53.6±.5	66.1±.2	48.5±.3	0.34±.00	18.0±.1	43.0±.1	56.6±.2	39.2±.1	263.0±0.9
SmoothAP	2.4	$\tau = 0.01$	30.8±.3	60.3±.2	**73.6**±.5	54.9±.3	0.40±.00	20.3±.2	46.5±.2	60.1±.2	42.3±.2	291.5±1.4
			VSRN									
Triplet loss	2.5	$\alpha = 0.2$	42.9±.4	74.3±.3	84.9±.4	67.4±.3	0.52±.00	33.5±.1	65.1±.1	77.1±.2	58.6±.1	377.8±1.2
Triplet loss SH	2.6	$\alpha = 0.2$	**48.9**±.6	**78.1**±.5	**87.4**±.2	71.4±.4	0.57±.01	**37.8**±.5	**68.1**±.5	**78.9**±.3	61.6±.4	**399.0**±2.3
NT-Xent	2.7	$\tau = 0.1$	37.9±.4	69.2±.2	80.7±.3	62.6±.1	0.47±.00	29.5±.1	61.0±.2	74.0±.2	54.6±.1	352.3±0.5
SmoothAP	2.8	$\tau = 0.01$	46.0±.6	76.1±.3	85.9±.3	69.4±.3	0.54±.00	33.8±.3	64.1±.1	76.0±.2	58.0±.2	382.0±1.1

Results. Based on the scores reported in Table 4.1, we have the following observations:

(i) Given a fixed method and default hyper-parameters for each loss function, the triplet loss SH results in the best evaluation scores, regardless of dataset, method or task.

(ii) Similar to (Faghri et al., 2018), we find that the triplet loss SH consistently outperforms the general triplet loss, which takes all the negative triplets in the batch into account that violate the margin constraint.

(iii) The NT-Xent loss consistently underperforms compared to the triplet loss SH. This is in contrast with findings in (Chen et al., 2020c), where the NT-Xent loss results in better down-stream evaluation performance on a (augmented image-to-image) representation learning task than the triplet loss SH. Although the ICR task has different (input) data modalities, the underlying learning object is the same for ICR and augmented image-to-image representation learning (i.e., contrasting positive and negative pairs).

(iv) Only for the VSE++ method on the i2t task, SmoothAP performs similarly to the triplet loss SH.

(v) SmoothAP does not outperform the triplet loss SH. This is in contrast with the findings in (Brown et al., 2020a), where SmoothAP does outperform triplet loss SH and other metric learning functions.

(vi) The method with the best recall@k score also has the highest mAP@k score.

Upshot. Based on our observations concerning Table 4.1, we conclude the following:

(i) The triplet loss SH should still be the *de facto* choice for optimizing ICR methods.

(ii) The promising results from the representation learning field that were obtained by using the NT-Xent loss (Chen et al., 2020c), do not generalize to the ICR task.

(iii) Optimizing an ICR method with a smooth approximation of a ranking metric (SmoothAP) does not result in better recall@k scores.

(iv) Optimizing an ICR method by using a pair-wise distance loss between the positive triplet and a semi-hard negative triplet still yields the best evaluation performance. For both methods VSE++ and VSRN, both the i2t and t2i tasks, and both datasets Flickr30k and MS-COCO.

4.4 A METHOD FOR ANALYZING THE BEHAVIOR OF LOSS FUNCTIONS

Next, we propose a method for analyzing the behavior of loss functions for ICR. The purpose is to compare loss functions and explain the difference in performance. If we compare the gradient w.r.t. $\mathbf{q}$ for the triplet loss and the triplet loss SH, the only difference is the number of triplets that the two loss functions take into account. If two models are optimized in exactly the same manner, except one model uses the triplet loss and the other uses triplet loss SH, the difference in performance can only be explained by the fact that the triplet loss takes all violating triplets into account. This means that the number of triplets (i.e., candidates) that contribute to the gradient directly relates to the evaluation performance of the model. The same reasoning applies to the NT-Xent and the SmoothAP loss. For example, the gradient w.r.t. $\mathbf{q}$ for the NT-Xent loss also has the form $\mathbf{v}^+ - \mathbf{v}^-$. The major difference between the two functions is that for the negative candidate the NT-Xent loss computes a weighted sum over all negatives to compute a representation of $\mathbf{v}^-$. Therefore, the difference in evaluation performance between the triplet loss SH and NT-Xent can only be explained by this weighted sum over all negatives. This sum can be turned into a count of negatives, i.e., how many negatives approximately contribute to this weighted sum, which can be related to the other losses. By counting the number of candidates that contribute to the gradient, we aim to get a better understanding of why a certain loss function performs better than others. The method we propose is called *counting contributing samples (COCOS)*.

First, we provide the form of the derivative of each loss function w.r.t. query $\mathbf{q}$. For each loss function the derivative is a sum over $\mathbf{v}^+ - \mathbf{v}^-$. Loss functions may weigh the positive and negative candidate(s) differently, and the number of candidates or triplets that are weighted may differ across loss functions.

4.4.1 Triplet loss and triplet loss SH

The gradient w.r.t. $\mathbf{q}$ for the triplet loss SH, $\mathcal{L}^{\mathbf{q}}_{TripletSH}$ is the difference between the representation of the positive and negative candidate:

$$\frac{\partial \mathcal{L}^{\mathbf{q}}_{TripletSH}}{\partial \mathbf{q}} = \begin{cases} \mathbf{v}^+ - \mathbf{v}^-, & \text{if } s^+ - s^- < \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{4.9a}$$

$$\frac{\partial \mathcal{L}^{\mathbf{q}}_{Triplet}}{\partial \mathbf{q}} = \sum_{\mathbf{v}^- \in \mathcal{N}_{\mathbf{q}}} \mathbb{1}\{s^+ - s^- < \alpha\} \left(\mathbf{v}^+ - \mathbf{v}^-\right). \tag{4.9b}$$

The gradient of triplet loss $\mathcal{L}^{\mathbf{q}}_{Triplet}$ (Eq. 4.9b) w.r.t. $\mathbf{q}$ has a similar form. However, there the gradient is a sum over all triplets that violate $s^+ + s^- < \alpha$, and not only the maximum violating one. Based on Eq. 4.9a we can see that a query $\mathbf{q}$ only has a non-zero gradient when $s^+ - s^- < \alpha$. If this is the case, the gradient always has the form $\mathbf{v}^+ - \mathbf{v}^-$, and this value is independent of the magnitude $s^+ - s^-$. For this reason, given a batch $\mathcal{B}$, the number of queries $\mathbf{q}$ that have a non-zero gradient is defined by:

$$C^{\mathcal{B}}_{TripletSH} = \sum_{\mathbf{q} \in \mathcal{B}} \mathbb{1}\{s^+ - s^- < \alpha\}, \tag{4.10}$$

where $s^+ = s^0 \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}$ and $s^- = \max\left(\mathcal{S}^{\mathbf{q}}_{\mathcal{N}}\right)$. We define $C^{\mathcal{B}}_{TripletSH}$ to be *the number of queries* $\mathbf{q}$ *that have a non-zero gradient given batch* $\mathcal{B}$.

As the triplet loss takes all the triplets into account that violate the distance margin α, we can count three things:

(i) Per query $\mathbf{q}$, we can count how many triplets $\mathbf{v}^+ - \mathbf{v}^-$ contribute to the gradient of $\mathbf{q}$. We define this as $C^{\mathbf{q}}_{Triplet} = \sum_{s^- \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathbb{1}\{s^+ - s^- < \alpha\}$.

(ii) Given the batch $\mathcal{B}$, we can count how many triplets contribute to the gradient over the entire training batch $\mathcal{B}$. We define this number as $C^{\mathcal{B}}_{Triplet} = \sum_{\mathbf{q} \in \mathcal{B}} C^{\mathbf{q}}_{Triplet}$.

(iii) Given the entire batch $\mathcal{B}$, we can count how many queries have a gradient value of zero (i.e., no violating triplets). This number is $C^0_{Triplet} = \sum_{\mathbf{q} \in \mathcal{B}} \mathbb{1}\{C^{\mathbf{q}}_{Triplet} = 0\}$.

4.4.2 NT-Xent loss

The gradient w.r.t. $\mathbf{q}$ for the NT-Xent loss is defined as (Chen et al., 2020c):

$$\frac{\partial \mathcal{L}^{\mathbf{q}}_{NT\text{-}Xent}}{\partial \mathbf{q}} = \left(1 - \frac{\exp(s^+/\tau)}{Z(\mathbf{q})}\right) \tau^{-1} \mathbf{v}^+ - \sum_{s^- \in \mathcal{S}^q_{\mathcal{N}}} \left(\frac{\exp(s^-/\tau)}{Z(\mathbf{q})}\right) \tau^{-1} \mathbf{v}^-, \tag{4.11}$$

where $Z(\mathbf{q}) = \sum_{s_i \in \mathcal{S}^{\mathbf{q}}_{\Omega}} \exp(s_i/\tau)$, a normalization constant depending on $\mathbf{q}$. The gradient w.r.t. $\mathbf{q}$ is the weighted difference of the positive candidate $\mathbf{v}^+$ and the weighted sum over all the negative candidates. The weight for each candidate is based on the similarity with the query, normalized by the sum of the similarities of all candidates. In contrast, for the triplet-loss (Eq. 4.9b) all candidates are weighted equally when they violate the margin constraint. The NT-Xent loss performs a natural form of (hard) negative weighting (Chen et al., 2020c). The more similar a negative sample is to the query, the higher the weight of this negative in the gradient computation. In principle, all the negatives and the positive candidate contribute to the gradient w.r.t. $\mathbf{q}$. In practice, most similarity scores $s^- \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}$ have a low value; so the weight of this negative candidate in the gradient computation will be close to 0.

To count the number of negative candidates that contribute to the gradient, we define a threshold value ϵ. If the weight of a negative candidate $\mathbf{v}^-$ is below ϵ, we assume that its contribution is negligible. All candidate vectors are normalized. Hence, there is no additional weighting effect by the magnitude of the vector. For the NT-Xent loss we define three terms: $C^{\mathbf{qv}^-}_{NTXent}$, $W^{\mathbf{qv}^-}_{NTXent}$ and $W^{\mathbf{qv}^+}_{NT-Xent}$:

(i) Given a query $\mathbf{q}$, $C^{\mathbf{qv}^-}_{NT-Xent}$ is the number of negative candidates $\mathbf{v}^-$ that contribute to the gradient w.r.t. $\mathbf{q}$: $C^{\mathbf{qv}^-}_{NT-Xent} = \sum_{s^- \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathbb{1}\{\exp(s^-/\tau)Z(\mathbf{q})^{-1} > \epsilon\}$. We count all the negative candidates $\mathbf{v}^-$ that have a normalized similarity score with the query higher than ϵ.

(ii) Given $C^{\mathbf{qv}^-}_{NT-Xent}$, we compute the sum of the weight values of the contributing negative canidates $\mathbf{v}^-$ as $W^{\mathbf{qv}^-}_{NT-Xent} = \sum_{s^- \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathbb{1}\{\exp(s^-/\tau)Z(\mathbf{q})^{-1} > \epsilon\} \exp(s^-/\tau)Z(\mathbf{q})^{-1}$.

(iii) We define $W^{\mathbf{qv}^+}_{NT-Xent} = \frac{1}{N}\sum_{\mathbf{q} \in \mathcal{B}}(1 - \exp(s^+/\tau)Z(\mathbf{q})^{-1})$, as the mean weight value of the positive candidates in batch $\mathcal{B}$.

We define the two extra terms, $W^{\mathbf{qv}^-}_{NT-Xent}$ and $W^{\mathbf{qv}^+}_{NT-Xent}$ because for the NT-Xent function, we have to count the candidates with a weight value above the threshold ϵ. This count on its own does not provide a good picture of the contribution of these candidates to the gradient. Therefore, we compute a mean value of those weight values as well, to provide insight into the number of the samples on which the gradient w.r.t. $\mathbf{q}$ is based.

4.4.3 SmoothAP loss

A full derivation of the gradient of SmoothAP w.r.t. $\mathbf{q}$ is provided in Appendix 4.B. We introduce $sim(D_{ij})$, the derivative of Eq. 4.7:

$$\frac{\partial AP_{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)^{-2} \left(\mathcal{R}\left(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}\right) \left(\sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j) \right) - \left(\mathcal{R}\left(i, \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}\right) - 1\right) \left(\sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j) \right) \right). \tag{4.12}$$

Given Eq. 4.12, it is less trivial to infer what the update w.r.t. $\mathbf{q}$ looks like in terms of positive candidates $\mathbf{v}_i$ and negative candidates $\mathbf{v}_j$. However, we can derive the following two properties:

(i) The lower a positive candidate $\mathbf{v}_i$ is in the total ranking, the less this candidate is taken into account for the gradient computation w.r.t. $\mathbf{q}$, due to the inverse quadratic term $\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{q}\right)^{-2}$. This is in line with optimizing the AP as a metric; positive candidates that are ranked low contribute less to the total AP score and therefore are less important to optimize.

(ii) Each triplet $\mathbf{v}_i - \mathbf{v}_j$ is weighted according to their difference in similarity score D_{ij}. If their difference in similarity score w.r.t. query $\mathbf{q}$ is relatively small (i.e., D_{ij} is close to zero), $sim(D_{ij})$ will have a high value due to the fact that $sim(D_{ij})$ is the derivative of the sigmoid function. Therefore, $sim(D_{ij})$ indicates how close the similarity score (with the query) of candidate $\mathbf{v}_i$ is compared to the similarity score of $\mathbf{v}_j$. This is in line with the SmoothAP loss because we use a sigmoid to approximate the step-function; only triplets of candidates that have a similar similarity score will contribute to the gradient.

We define a threshold value ϵ again. If the value of $sim(D_{ij})$ is lower than the threshold value, we consider the contribution of this triplet to be negligible. We have to take into account that all triplets are also weighted by $\mathcal{R}(i, \mathcal{S}_{\Omega}^{\mathbf{q}})^{-2}$, which is always lower than or equal to 1. We can define $C_{Smooth}^{\mathbf{q}}$, which *is the*

number of triplets $\mathbf{v}^+ - \mathbf{v}^-$ *that contribute to the gradient w.r.t.* $\mathbf{q}$, for SmoothAP as follows:

$$C^{\mathbf{q}}_{Smooth} = \frac{1}{|\mathcal{S}^{\mathbf{q}}_{\mathcal{P}}|} \sum_{i \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}} \left(\sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}} \mathbb{1}\left\{ \frac{sim(D_{ij})}{\mathcal{R}(i, \mathcal{S}^{\mathbf{q}}_{\Omega})^2} > \epsilon \right\} + \sum_{j \in \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}, j \neq i} \mathbb{1}\left\{ \frac{sim(D_{ij})}{\mathcal{R}(i, \mathcal{S}^{\mathbf{q}}_{\Omega})^2} > \epsilon \right\} \right). \tag{4.13}$$

Similar to (Brown et al., 2020a), we use $sim(D_{ij})$ in combination with a threshold value ϵ to indicate which samples have a non-zero gradient in the training batch. We ignore the terms $\mathcal{R}\left(i, \mathcal{S}^{\mathbf{q}}_{\mathcal{P}}\right)$ and $1 - \mathcal{R}\left(i, \mathcal{S}^{\mathbf{q}}_{\mathcal{N}}\right)$ for this gradient computation. We also count all queries $\mathbf{q}$ within batch $\mathcal{B}$ that do not have a gradient value. We define this number as $C^0_{Smooth} = \sum_{\mathbf{q} \in \mathcal{B}} \mathbb{1}\{C^{\mathbf{q}}_{Smooth} = 0\}$. This completes the definition of COCOS: for every loss function that we consider, it counts the number of candidates that contribute to the gradient w.r.t. $\mathbf{q}$.

4.5 ANALYZING THE BEHAVIOR OF LOSS FUNCTIONS FOR ICR

4.5.1 Experimental setup

To use COCOS, we introduce the following experimental setup. For each loss function, we take the checkpoint of one of the five optimized models. We refer to this checkpoint as *the optimal convergence point* for this loss function. This is not the point with the lowest loss value, but the model checkpoint that results in the highest evaluation scores on the validation set. We freeze all model parameters and do not apply dropout. We iterate over the entire training set by sampling random batches $\mathcal{B}$ (with batch size $|\mathcal{B}| = 128$, similar to the training set-up). For each batch, we compute the COCOS and weight values defined in Section 4.4. We report the mean value and standard deviation over the entire training set for both VSE++ and VSRN, for both datasets and for each loss function. The only hyper-parameter for this experiment is ϵ. We use $\epsilon = 0.01$ for both the NT-Xent and SmoothAP loss.

Table 4.2: COCOS w.r.t. query $\mathbf{q}$, for the triplet loss and the triplet loss SH.

				i2t			t2i		
			#	$\mathcal{C}^{\mathbf{q}}$	$\mathcal{C}^{\mathcal{B}}$	$\mathcal{C}^{0}$	$\mathcal{C}^{\mathbf{q}}$	$\mathcal{C}^{\mathcal{B}}$	$\mathcal{C}^{0}$
Flickr30k	VSE++	Triplet loss	1.1	6.79±0.83	768.92±96.87	14.78±3.52	6.11±0.75	774.67±98.05	1.14±1.22
		Triplet loss SH	1.2	1±0.0	98.74±4.83	29.23±4.81	1±0.0	98.22±4.66	29.75±4.62
	VSRN	Triplet loss	1.5	1.39±0.12	60.96±10.30	84.29±5.80	1.28±0.10	61.21±10.01	80.15±6.35
		Triplet loss SH	1.6	1±0.0	45.59±5.93	82.39±5.92	1±0.0	44.98±5.70	82.99±5.70
MS-COCO	VSE++	Triplet loss	2.1	3.51±0.49	353.82±52.71	27.09±4.60	2.94±0.36	341.64±50.80	12.24±4.92
		Triplet loss SH	2.2	1 ±0.0	88.17±5.25	39.82±5.24	1±0.0	87.24±5.34	40.75±5.33
	VSRN	Triplet loss	2.5	1.21±0.13	29.88±7.46	103.33±5.22	1.15±0.10	30.25±7.49	101.70±5.58
		Triplet loss SH	2.6	1±0.0	33.24±5.39	94.73±5.45	1±0.0	32.90±5.35	95.08±5.4

4.5.2 Experimental outcomes

For each of the loss functions that we consider, we analyze its performance using COCOS.

Triplet loss. Our goal is not to show that the triplet loss SH outperforms the triplet loss, which has already been shown (Faghri et al., 2018), but to explain this behavior based on COCOS w.r.t. $\mathbf{q}$ and also relate this to the NT-Xent and SmoothAP loss.

Based on Table 4.1 (row 1.1/1.2 and 1.5/1.6, row 2.1/2.2 and 2.5/2.6) it is clear that the triplet loss SH always outperforms the general triplet loss with a large margin. If we look at Table 4.2, row 1.1/1.2 and 2.1/2.2, respectively, there is a clear relation between $\mathcal{C}^{\mathbf{q}}$ and the final evaluation score for the VSE++ model for both sub-tasks i2t and t2i (Table 4.1). $\mathcal{C}^{\mathbf{q}}_{Triplet}$ and $\mathcal{C}^{\mathcal{B}}_{Triplet}$ are both much greater than $\mathcal{C}^{\mathbf{q}}_{TripletSH}$ and $\mathcal{C}^{\mathcal{B}}_{TripletSH}$, for both dataset and both the the i2t and t2i task. When multiple negatives with a small margin violation are combined into a gradient, the gradient is dominated by easy or non-informative negative samples, which results in convergence of the model into a sub-optimal point (Faghri et al., 2018). Clearly, the loss function with the lowest evaluation score takes into account the most negatives when computing the gradient w.r.t. $\mathbf{q}$. Based on (Faghri et al., 2018) and the COCOS results in Table 4.2 we conclude that, at the optimal convergence point, the triplet loss takes too many negatives into account (i.e., too many triplets still violate the margin constraint), leading to lower evaluation scores.

For VSRN the relation between $\mathcal{C}^{\mathbf{q}}_{Triplet}$, $\mathcal{C}^{\mathbf{q}}_{TripletSH}$ and the final evaluation score is less clear. If we look at Table 4.2, row 1.5/1.6 and 2.5/2.6, respectively, we see that $C^{\mathbf{q}}_{Triplet} \approx C^{\mathbf{q}}_{TripletSH} = 1$. This means that at the optimal

Table 4.3: COCOS w.r.t. query $\mathbf{q}$, for the NT-Xent loss (Chen et al., 2020c).

			i2t			t2i		
		#	$C^{\mathbf{qv}^-}_{NT-Xent}$	$W^{\mathbf{qv}^-}_{NT-Xent}$	$W^{\mathbf{qv}^+}_{NT-Xent}$	$C^{\mathbf{qv}^-}_{NT-Xent}$	$W^{\mathbf{qv}^-}_{NT-Xent}$	$W^{\mathbf{qv}^+}_{NT-Xent}$
Flickr30k	VSE++	1.3	9.88±0.51	0.42±0.02	0.56±0.02	9.65±0.51	0.42±0.02	0.56±0.02
	VSRN	1.7	2.45±0.23	0.13±0.02	0.20±0.02	2.46±0.23	0.13±0.02	0.20±0.02
MS-COCO	VSE++	2.3	5.59±0.40	0.36±0.02	0.46±0.02	5.33±0.38	0.36±0.02	0.46±0.02
	VSRN	2.7	1.10±0.14	0.10±0.02	0.14±0.02	1.11±0.14	0.09±0.02	0.14±0.02

convergence point, for VSRN, the triplet loss and the triplet loss SH (approximately) are similar to each other and both functions only take one negative triplet into account when computing the gradient w.r.t. $\mathbf{q}$. Thus, both functions should result in approximately the same gradient value while the triplet loss SH still outperforms the triplet loss with a large margin. This can be explained as follows: At the start of training, for each query $\mathbf{q}$ (almost) all triplets violate the margin constraint (because all candidate representations are random). Therefore, the gradient(s) computation w.r.t. $\mathbf{q}$ for the triplet loss is based on all triplets in the batch and therefore this gradient is dominated by a majority of non-informative samples at the beginning of the training, which leads to convergence at a sub-optimal point.

NT-Xent. Based on Table 4.3, we can see that $C^{\mathbf{qv}^-}_{NT-Xent}$ is higher than 1 for both VSE++ and VSRN, for i2t and t2i, on both datasets. If we relate the evaluation performances of the NT-Xent loss (row 1.3, 1.7, 2.3, 2.7) to the triplet loss SH (row 1.2, 1.6, 2.2, 2.6) in Table 4.1, we can see that the triplet loss SH consistently outperforms the NT-Xent loss, regardless of the method, dataset or sub-task. We therefore can conclude that taking only the most violating negative into account when computing the gradient w.r.t. $\mathbf{q}$ results in better evaluation performances than computing a weighted sum over all negative candidates. We can apply the same reasoning used to explain the performance difference between the triplet loss and triplet loss SH. The gradient w.r.t. $\mathbf{q}$ for the NT-Xent is dominated by too many non-informative negatives, which have a weight value bigger than ϵ.

Looking at Table 4.1, we see that NT-Xent loss outperforms the triplet loss for the VSE++ method (1.3/1.1 and 2.3/2.1) while taking more negative samples into account when computing the gradient (based on our definition of COCOS). This in contrast with the previous observation for the triplet loss of the more

Table 4.4: COCOS w.r.t. query $\mathbf{q}$, for the SmoothAP (Brown et al., 2020a) loss.

			i2t		t2i	
		#	$C^{\mathbf{q}}_{SmoothAP}$	$C^{0}_{SmoothAP}$	$C^{\mathbf{q}}_{SmoothAP}$	$C^{0}_{SmoothAP}$
Flickr30k	VSE++	1.4	1.27±0.06	2.15±1.51	1.47±0.83	636.72±18.72
	VSRN	1.8	2.33±0.07	0.00±0.00	1.62±0.95	636.49±18.65
MS-COCO	VSE++	2.4	1.48±0.07	0.80±0.90	1.41±0.74	637.10±20.28
	VSRN	2.8	1.67±0.07	0.14±0.37	1.42±0.76	637.23±20.35

(non-informative) samples a loss function takes into account when computing the gradient w.r.t. $\mathbf{q}$, the lower the evaluation score. Solely counting the number of negative examples that contribute to the gradient does not provide the full picture for the NT-Xent loss; the weight value of each individual sample (including the positive) plays a more important role than initially was assumed. We have tried different values for ϵ, with little impact.

SmoothAP. The observations in Table 4.4 are in line with the observations in Table 4.2 and the evaluation performance in Table 4.1. At the optimal convergence point SmoothAP takes approximately one triplet into account when computing the gradient w.r.t. $\mathbf{q}$, which results in close-to or similar performances as the triplet loss SH. We also observe the following: the only experiment where the triplet loss SH outperforms SmoothAP with a large margin (Table 4.1, row 1.5 and 1.8), is also the experiment where the SmoothAP function takes the highest number of negatives into account when computing the gradient w.r.t. $\mathbf{q}$ (Table 4.4, row 1.8). This supports the general observation that the more samples that contribute to the gradient, the lower the final evaluation score.

For the t2i task, we also see that $C^{0}_{SmoothAP}$ is almost as big as the number of samples (640=$(k = 5) \times (|\mathcal{B}| = 128)$) in the candidate set, for both datasets and methods. Hence, barely any query has a gradient value anymore at the optimal convergence point. However, this is not the case for the i2t task. We conclude that optimizing a ranking metric (i.e., AP) with only one positive candidate (as is the case for the t2i task), might be too easy to optimize and could result in over-fitting. Therefore, it is not useful to optimize a ranking task like ICR with a ranking-based loss function when there is only one positive candidate per query, which is the case for the i2t task. For the i2t task, however, there are

barely any queries without a gradient value; here we have k positive candidates per query.

Upshot. In summary, the main insights from this section are as follows:

(i) It is important to focus on only one (or a limited) number of (hard) negatives per query during the entire training for the gradient computation, so as to prevent the gradient from being dominated by non-informative or easy negative samples.

(ii) Weighting each negative candidate by its score (as is done in NT-Xent) as opposed to weighting all negatives equally (as is done in the triplet loss) can be beneficial for the gradient computation and therefore for the final evaluation score. However, this weighted sum of negatives does not result in the fact that the NT-Xent loss outperforms the triplet loss SH, which implies that the gradient computation for the NT-Xent is still based on too many non-informative samples.

4.6 DISCUSSION & CONCLUSION

In this chapter, we have examined three loss functions from the metric learning field to analyze if the promising results obtained in metric learning generalize to the image-caption retrieval task. In contrast with the findings from metric learning, we find that the triplet loss with semi-hard negative mining still outperforms the NT-Xent and SmoothAP loss. Hence, the triplet loss should still be the de facto choice as a loss function for ICR; results from metric learning do not generalize directly to ICR. Therefore, we answered the third research question of this thesis: lessons from metric learning do not generalize one-on-one to ICR. To gain a better understanding of why a loss function results in better performance than others, we have introduced the notion of counting contributing samples (COCOS). We have shown that the best performing loss function only focuses on one (hard) negative sample when computing the gradient w.r.t. the query and therefore results in the most informative gradient. COCOS suggests that the underperforming loss functions take too many (non-informative) negatives into account, and therefore converge to a sub-optimal point.

The definition of COCOS uses a threshold value. The idea that a candidate contributes to the gradient if its weight value is above a certain threshold is in-

sightful but does not provide the complete picture of how strong the influence of this sample is. We encourage two directions for future work:

(i) work on more sophisticated methods to determine the influence of (the number of) samples on the gradient w.r.t. a query.

(ii) Design new loss functions for the ICR task by taking the lessons from COCOS into account, i.e., loss functions that only take one, or a limited number of, hard negative(s) into account.

Additionally, we want to investigate if our findings generalize to fields such as dense passage retrieval (Karpukhin et al., 2020). Dense passage retrieval methods are also mainly optimized by using two data encoders (Karpukhin et al., 2020; Khattab and Zaharia, 2020), for the query and for documents, and the main learning objective is contrasting positive and negative candidates with a query (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Chen et al., 2021c; Formal et al., 2021; Gao et al., 2021; Zhan et al., 2021), similar to ICR.

In the next chapter, we continue with the ICR evaluation task for contrastive image-text representation learning methods. We first show that contrastive image-text methods that are optimized with the InfoNCE loss (i.e., NT-Xent) are prone to predictive feature suppression. We then introduce a method to reduce predictive feature suppression for resource-constrained ICR methods.

Chapter Appendix

The appendix of this chapter has three sections, one devoted to the notation and variables used throughout this chapter (Appendix 4.A), one to the derivation of the gradient of SmoothAP w.r.t. $\mathbf{q}$ (Appendix 4.B), and one devoted to reproducibility (Appendix 4.C).

4.A NOTATION AND VARIABLES

Table 4.A.1: Overview of the notation and variables used throughout Chapter 4.

Symbol	Explanation
$\mathcal{D}$	Dataset $\mathcal{D}$ consisting of N image-caption tuples.
$\mathbf{x}_I^i$	Input image from tuple $i \in \mathcal{D}$
$\mathbf{x}_{C_k}^i$	Input caption j, where $1 \leq j \leq k$ (k is number of captions per tuple), from tuple $i \in \mathcal{D}$.
$\mathbf{q}$	Latent representation of the query, either an image or a caption.
$\mathbf{v}_i$	Candidate i in candidate set Ω.
$\mathbf{v}^+$	A matching candidate (i.e., positive) given query $\mathbf{q}$, either an image or a caption.
$\mathbf{v}^-$	A non-matching candidate (i.e., negative) given query $\mathbf{q}$, either an image or a caption.
Ω	Set of all candidates representations $\mathbf{v}_i$ to be ranked w.r.t. query $\mathbf{q}$.
$\mathcal{N}_\mathbf{q}$	Set of all non-matching candidates $\mathbf{v}^-$ w.r.t. query $\mathbf{q}$.
$\mathcal{P}_\mathbf{q}$	Set of all matching candidates $\mathbf{v}^+$ w.r.t. query $\mathbf{q}$.
$\mathcal{S}_\Omega^\mathbf{q}$	Set with similarity scores for each $\mathbf{v}_i \in \Omega$ w.r.t. query $\mathbf{q}$.
$\mathcal{S}_\mathcal{P}^\mathbf{q}$	Set with similarity scores for each $\mathbf{v}_i \in \mathcal{P}_\mathbf{q}$ w.r.t. query $\mathbf{q}$.
	Continued on next page

Table 4.A.1 – continued from previous page

Symbol	Explanation
$\mathcal{S}_{\mathcal{N}}^{\mathbf{q}}$	Set with similarity scores for each $\mathbf{v}_i \in \mathcal{N}_{\mathbf{q}}$ w.r.t. query $\mathbf{q}$.
$\mathcal{B}$	Training batch with image-caption pairs.
α	Margin parameter for the triplet loss.
τ	Temperature parameter to scale the logist (i.e., cosine similarity) for the NT-Xent loss or SmoothAP scores.
$\mathcal{L}_{Triplet}$	Triplet loss.
$\mathcal{L}_{Triplet}^{\mathbf{q}}$	Triplet loss w.r.t query $\mathbf{q}$.
$\mathcal{L}_{NT-Xent}$	NT-Xent loss.
$\mathcal{L}_{AP}$	SmoothAP loss.
$\mathcal{C}^{\mathcal{B}}$	Number of queries $\mathbf{q}$ that have a non-zero gradient given batch $\mathcal{B}$, for a given loss function.
$\mathcal{C}^{0}$	Number of queries $\mathbf{q}$ that have a zero gradient given batch $\mathcal{B}$, for a given loss function.
$\mathcal{C}^{\mathbf{q}}$	Numbers of triplets that contribute to the gradient w.r.t. q, given a loss function.

4.B DERIVATIVE OF THE GRADIENT OF SMOOTHAP W.R.T. q

4.B.1 Explanation of SmoothAP

The average precision metric represents the area under the precision-recall curve. Average precision is a discrete metric and therefore can not be used directly as a loss function for optimizing retrieval methods. The main intuition behind the SmoothAP (Brown et al., 2020a) is to have a smooth, and therefore differentiable, approximation of the average precision metric. Using the notation introduced in Section 4.2, the average precision metric is defined as follows:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{\mathcal{R}(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}})}{\mathcal{R}(i, \mathcal{S}_{\Omega}^{\mathbf{q}})}, \tag{4.14}$$

where $\mathcal{R}(i, \mathcal{S})$ is defined as:

$$\mathcal{R}(i, \mathcal{S}) = 1 + \sum_{j \in \mathcal{S}, i \neq j} \mathbb{1}\{s_i - s_j < 0\}. \tag{4.15}$$

$\mathcal{R}(i, \mathcal{S})$ returns the rank of candidate i in a ranking over a set with candidates $\mathcal{S}$, given query $\mathbf{q}$. s_i is the similarity score between the query $\mathbf{q}$ and candidate $\mathbf{v}_i$. Similarly, s_j is similarity score between the query $\mathbf{q}$ and candidate $\mathbf{v}_j$. If $s_i - s_j$ is lower than 0, this indicates that candidate j is ranked higher than candidate i. By counting how many times $\mathbb{1}\{s_i - s_j < 0\}$ is true, the ranking of candidate i can be determined.

To simplify the computation and notation, we can introduce a matrix D, where D is defined as:

$$D = \begin{bmatrix} s_1 & \dots & s_m \\ \vdots & \ddots & \vdots \\ s_1 & \dots & s_m \end{bmatrix} - \begin{bmatrix} s_1 & \dots & s_1 \\ \vdots & \ddots & \vdots \\ s_m & \dots & s_m \end{bmatrix}, \tag{4.16}$$

where $D_{ij} = s_i - s_j$. In this case we have a candidate set $\mathcal{S}$ with m candidates. By using matrix D, we can rewrite Eq. 4.15 as follows:

$$\mathcal{R}(i, \mathcal{S}) = 1 + \sum_{j \in \mathcal{S}, i \neq j} \mathbb{1}\{D_{ij} > 0\}. \tag{4.17}$$

To make $\mathcal{R}(i, \mathcal{S})$, and thereby $AP_{\mathbf{q}}$, differentiable, the indicator function $\mathbb{1}$ is replaced by the smooth sigmoid function $\mathcal{G}(\cdot, \tau)$.

4.B.2 *Derivative of the gradient of SmoothAP w.r.t. q*

In this section, we give an analyses and derivation of the gradient of SmoothAP (Brown et al., 2020a) w.r.t. query $\mathbf{q}$. We start with Eq. 4.18, the definition of SmoothAP:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij}; \tau)}{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij}; \tau) + \sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} \mathcal{G}(D_{ij}; \tau)}. \tag{4.18}$$

Here, $\mathcal{G}$ is a smooth approximation of an indicator/step function:

$$\mathcal{G}\left(f(x); \tau\right) = \frac{1}{1 + e^{-\frac{f(x)}{\tau}}}. \tag{4.19}$$

The derivative of $\mathcal{G}$ w.r.t. a function $f(x)$ has the following form:

$$\frac{\partial \mathcal{G}\left(f(x);\tau\right)}{\partial x} = \mathcal{G}\left(f(x);\tau\right)\left(1-\mathcal{G}\left(f(x);\tau\right)\right)\frac{1}{\tau}\frac{\partial f(x)}{\partial x}. \tag{4.20}$$

Note that $f(x)$ in the case of SmoothAP is D_{ij}:

$$D_{ij} = s_i - s_j = \mathbf{q}\mathbf{v}_i - \mathbf{q}\mathbf{v}_j, \tag{4.21}$$

where both $\mathbf{v}_i$, $\mathbf{v}_j$ and $\mathbf{q}$ are normalized on the unit-sphere. The gradient of D_{ij} w.r.t. query $\mathbf{q}$ has the following form:

$$\frac{\partial D_{ij}}{\partial \mathbf{q}} = \mathbf{v}_i - \mathbf{v}_j. \tag{4.22}$$

If we plug in D_{ij} into Eq. 4.20 and take the gradient w.r.t. query $\mathbf{q}$, we get

$$\begin{aligned}\frac{\partial \mathcal{G}\left(D_{ij};\tau\right)}{\partial \mathbf{q}} &= \mathcal{G}\left(D_{ij};\tau\right)\left(1-\mathcal{G}\left(D_{ij};\tau\right)\right)\frac{1}{\tau}(\mathbf{v}_i-\mathbf{v}_j)\\ &= sim\left(D_{ij},\tau\right)\left(\mathbf{v}_i-\mathbf{v}_j\right),\end{aligned} \tag{4.23}$$

where $sim(D_{ij},\tau)$ is a function that gives an indication of how close the similarity scores are of candidate i and j are w.r.t. query $\mathbf{q}$, scaled by τ:

$$sim(D_{ij},\tau) = \mathcal{G}(D_{ij};\tau)(1-\mathcal{G}(D_{ij};\tau))\frac{1}{\tau}. \tag{4.24}$$

Now we define $\mathcal{R}\left(i,\mathcal{S}_\Omega^\mathbf{q}\right)$ and $\mathcal{R}\left(i,\mathcal{S}_\mathcal{P}^\mathbf{q}\right)$. $\mathcal{R}\left(i,\mathcal{S}_\Omega^\mathbf{q}\right)$ gives the ranking of candidate i within the full candidate set $\mathcal{S}_\Omega^\mathbf{q}$. $\mathcal{R}\left(i,\mathcal{S}_\mathcal{P}^\mathbf{q}\right)$. gives the rank of candidate i within the positive candidate set $\mathcal{S}_\mathcal{P}^\mathbf{q}$:

$$\mathcal{R}\left(i,\mathcal{S}_\Omega^\mathbf{q}\right) = \left(\overbrace{1+\sum_{j\in\mathcal{S}_\mathcal{P}^\mathbf{q}, j\neq i}\mathcal{G}(D_{ij};\tau)}^{A} + \overbrace{\sum_{j\in\mathcal{S}_\mathcal{N}^\mathbf{q}}\mathcal{G}(D_{ij};\tau)}^{C}\right) \tag{4.25}$$

$$\mathcal{R}\left(i,\mathcal{S}_\mathcal{P}^\mathbf{q}\right) = \left(\overbrace{1+\sum_{j\in\mathcal{S}_\mathcal{P}^\mathbf{q}, j\neq i}\mathcal{G}(D_{ij};\tau)}^{A}\right). \tag{4.26}$$

The gradient of $\mathcal{R}\left(i,\mathcal{S}_\Omega^\mathbf{q}\right)$ w.r.t. to $\mathbf{q}$ has the following form:

$$\frac{\partial \mathcal{R}\left(i,\mathcal{S}_\Omega^\mathbf{q}\right)}{\partial \mathbf{q}} = \left(\overbrace{\sum_{j\in\mathcal{S}_\mathcal{P}^\mathbf{q}, j\neq i} sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)}^{B} + \overbrace{\sum_{j\in\mathcal{S}_\mathcal{N}^\mathbf{q}} sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)}^{D}\right). \tag{4.27}$$

Using all the definitions above, we can write the full gradient of $AP_{\mathbf{q}}$ w.r.t. $\mathbf{q}$:

$$\frac{\partial AP_{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{|\mathcal{S}_P^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{\mathcal{R}\left(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}\right) \frac{\partial \mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)}{\partial \mathbf{q}} - \mathcal{R}\left(i, \mathcal{S}_{\Omega}^{q}\right) \left(\sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)\right)}{\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)^2} \tag{4.28}$$

$$= \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{1}{\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)^2} \left(\left(\overbrace{\mathcal{R}\left(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}\right)}^{A} \overbrace{\frac{\partial \mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)}{\partial \mathbf{q}}}^{B+D} \right) - \left(\overbrace{\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)}^{A+C} \left(\overbrace{\sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)}^{D} \right) \right) \right). \tag{4.29}$$

When looking at Eq. 4.29, it becomes clear that we have a function in the following form $A(B+D) - (A+C)B$. This can be rewritten to: $AB + AD - AB - CB = AD - CB$. If we apply this to Eq. 4.29, we end up with the following form:

$$\frac{\partial AP_{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{1}{\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)^2} \left(\overbrace{\mathcal{R}\left(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}\right)}^{A} \left(\overbrace{\sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)}^{D} \right) - \overbrace{\sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} \mathcal{G}(D_{ij}; \tau)}^{C} \left(\overbrace{\sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)}^{B} \right) \right) \tag{4.30}$$

$$= \frac{1}{|\mathcal{S}_P^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{1}{\mathcal{R}\left(i, \mathcal{S}_{\Omega}^{\mathbf{q}}\right)^2} \left(\mathcal{R}\left(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}\right) \left(\sum_{j \in \mathcal{S}_{\mathcal{N}}^{q}} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j) \right) - \left(\mathcal{R}\left(i, \mathcal{S}_{\mathcal{N}}^{q}\right) - 1\right) \left(\sum_{j \in \mathcal{S}_{P}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j) \right) \right). \tag{4.31}$$

4.C REPRODUCIBILITY

4.c.1 VSE++

The VSE++ (Faghri et al., 2018) is an ICR method that uses two encoders that do not share parameters: an image and a caption encoder. For the image encoder, two CNN networks have been used in (Faghri et al., 2018): ResNet-152 (He et al., 2016) and VGG19 (Simonyan and Zisserman, 2015), where ResNet-152 yields the best evaluation performances. To reduce the computation time of the training process, we have decided to use ResNet-50 instead of ResNet-152. Some preliminary experiments have shown that the differences in evaluation scores between ResNet-152 and 50 are relatively small, while ResNet-50 is faster to optimize. The ResNet network functions as a so-called backbone or feature extractor. On top of the ResNet network, a fully-connected layer is placed to map the extracted features to a multi-modal latent space. Only the weights of this fully-connected layer are optimized during training.

The caption encoder consists of a unidirectional GRU (Cho et al., 2014a) encoder. The word embeddings for the text encoder are trained end-to-end (from scratch) with the rest of the text encoder. The output of the last encoding step is the representation of the input caption. The output representations of both encoders are normalized on the unit sphere.

4.c.2 VSRN

VSRN (Li et al., 2019a) also consists of a separate text and image encoder. For the image encoder, VSRN uses pre-computed features as input. These features have been generated by a Faster R-CNN (Ren et al., 2015) model, which uses ResNet-101 (He et al., 2016) as a backbone, trained on the Visual Genomes dataset (Krishna et al., 2016). The feature map of the last convolutional layer serves as input representation for the next layer, each vector in this feature map represents a region in the input image. Next, a GCN (Kipf and Welling, 2017) is used to enhance the input feature vectors with relation information between each region in the input image. Finally, a GRU is used to compute the global representation of the different region vectors. This is done by feeding the region representations one by one into the GRU encoder as a sequence. VSRN uses the same text encoder as VSE++.

To generate an extra training signal, a caption generator is added to the training process. This generator is optimized to reconstruct the input caption based on the visual region feature representations. We have decided to remove this caption decoder from the learning algorithm. The reason for this is that we focus on ICR only and we want to exclude any additional learning signal from the training process.

4.c.3 Implementation and optimization details

Both VSE++ and VSRN are optimized for 30 epochs on the same datasets (Lin et al., 2014; Young et al., 2014). For VSE++, after 30 epochs of training, 15 epochs of additional fine-tuning are applied where the backbone of the image encoder is also optimized. We do not apply this additional fine-tuning step due to the following two reasons:

(i) We want to optimize VSE++ and VSRN for the same number of epochs, and

(ii) Our goal is not to have the best performing model but rather to evaluate the impact of a loss function.

Therefore, the weights of the feature extractor for the VSE++ image encoder are frozen during the entire training process in this work. All the other remaining implementation details and hyper-parameters in this work are similar to (Faghri et al., 2018; Li et al., 2019a).

5

Reducing Predictive Feature Suppression

In Chapter 4, we questioned the generalizability of contrastive losses (i.e., metric learning functions) to the ICR task. In this chapter,[1] we continue our investigation by taking a closer look at one of the contrastive losses we examined in Chapter 4: the NT-Xent loss (Chen et al., 2020c). However, from this chapter onwards, we will use the more widely used name for the NT-Xent loss, namely InfoNCE (van den Oord et al., 2018). Robinson et al. (2021) show that contrastive losses are prone to *predictive feature suppression* and that the representations learned by using such losses mainly depend on the difficulty of the discrimination task during training. In resource-constrained training setups (i.e., when either the amount of training data or the compute budget is limited), it is unlikely that the scale of the training is sufficient to make the learning problem difficult enough such that the InfoNCE loss guides the encoders to extract all predictive features in the input data. This leads us to the fourth research question of this thesis:

Research Question 4: *Can we reduce predictive feature suppression for resource-constrained contrastive image-text representation learning?*

To answer this research question, we introduce *latent target decoding (LTD)*. LTD is a non-auto-regressive reconstruction objective, which can be combined with a contrastive loss, that reconstructs the input caption in the latent space of a general-purpose sentence encoder. Instead of implementing LTD as an additional loss function, we propose to implement LTD as an optimization constraint. We show that constrained-based LTD consistently outperforms (given an evaluation metric) baseline ICR methods that are solely trained with a contrastive loss. These findings indicate that the representations learned by com-

1 This chapter is based on (Bleeker et al., 2023b).

bining LTD with a contrastive loss contain more predictive features of the input data that are relevant for a down-stream evaluation task.

5.1 INTRODUCTION

Image-caption retrieval (ICR) is the task of using an image or a caption as a query and ranking a set of candidate items in the other modality. Both the images and captions are mapped into a shared latent space by two encoders, which correspond to the two modalities. These encoders usually do not share parameters and are typically optimized with a contrastive loss function (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Wang et al., 2019; Chen et al., 2020a; Liu et al., 2020; Messina et al., 2020a; Messina et al., 2020b; Wang et al., 2020; Diao et al., 2021; Jia et al., 2021; Yu et al., 2021b). How well an ICR method generalizes beyond the specific training data depends on the features that the method has learned during training. The contrastive loss explicitly learns the similarity between positive (matching) candidates, while pushing away negative (non-matching) candidates during training. In the ideal situation, the contrastive objective optimizes the image and caption encoder such that both encoders extract all relevant information from the caption and image that is needed for matching the positive candidates during evaluation. However, it is not defined upfront what information is needed during evaluation for retrieving the correct item among a set of candidates.

Predictive feature suppression. Hermann and Lampinen (2020) show that, in the presence of two *predictive features* that redundantly predict the output label of the input data, a deep neural model preferentially represents one of the two predictive features while the other feature is suppressed. In this chapter, we define *predictive feature suppression* for ICR as the suppression of features by an encoder network during training that would be useful to correctly predict the match between a query and the positive candidate at inference time. For contrastive training tasks, the features that are relevant for matching the query with the positive candidate (i.e., the predictive features) mainly depend on the negative candidates in the training batch. Only optimizing the contrastive InfoNCE loss does not guarantee avoidance of *shortcut features* that suppress certain (predictive) input features and the learned features mainly depend on the difficulty of the discrimination task (Robinson et al., 2021). Especially in a

resource-constrained training setup, it is likely that the majority of the input features in the caption and image are redundant for learning the similarity between matching images and captions, due to the limited number of negative samples available to contrast with. The contrastive optimization objective is easy to solve by only using a small subset of the predictive input features of the captions and images. Suppressing predictive features during training is an undesirable side-effect of contrastive representation learning in a resource-constrained training setup, since some of these features might be needed during evaluation to retrieve the matching candidate. In Figure 5.1, we provide a visual example of predictive feature suppression in a resource-constrained contrastive image-text matching setup.

How to prevent predictive feature suppression. To increase the difficulty of a contrastive discrimination task, one can increase the batch size during training in order to increase the probability of having difficult in-batch negative samples (Chen et al., 2020c; Qu et al., 2021). It is, therefore, not surprising that most progress on the two widely used ICR benchmark evaluation sets, Flickr30k (Flickr30k, Young et al., 2014) and MS-COCO Captions (MS-COCO, Lin et al., 2014), has recently been made by using large-scale image-text matching training, mainly in combination with transformer network architectures (Jia et al., 2021; Yuan et al., 2021). Using more data and larger model architectures improves performance but comes with a significant extra computational cost, both in terms of data needed for training and the number of parameters that need to be optimized.

The two benchmark datasets for ICR, the Flickr30k and MS-COCO datasets, are relatively small in terms of training samples compared to the training data of state-of-the-art pre-trained ICR or image-text matching methods (Jia et al., 2021; Yuan et al., 2021). When an ICR method is trained from scratch using these benchmark datasets only, for example, in a resource-constrained training setup, scaling up the size of a batch is not a feasible solution to increase performance, due to the limited data size of Flickr30k and MS-COCO or due to the lack of computational resources. Hence, it is important to develop algorithms that can improve the effectiveness of ICR methods in a *resource-constrained* training setup, without relying on more data and more compute to achieve this.

A method to increase the difficulty of the contrastive objective that does not rely on the size of the dataset to reduce predictive feature suppression is to directly mine *hard* negative examples for each query over the entire dataset,

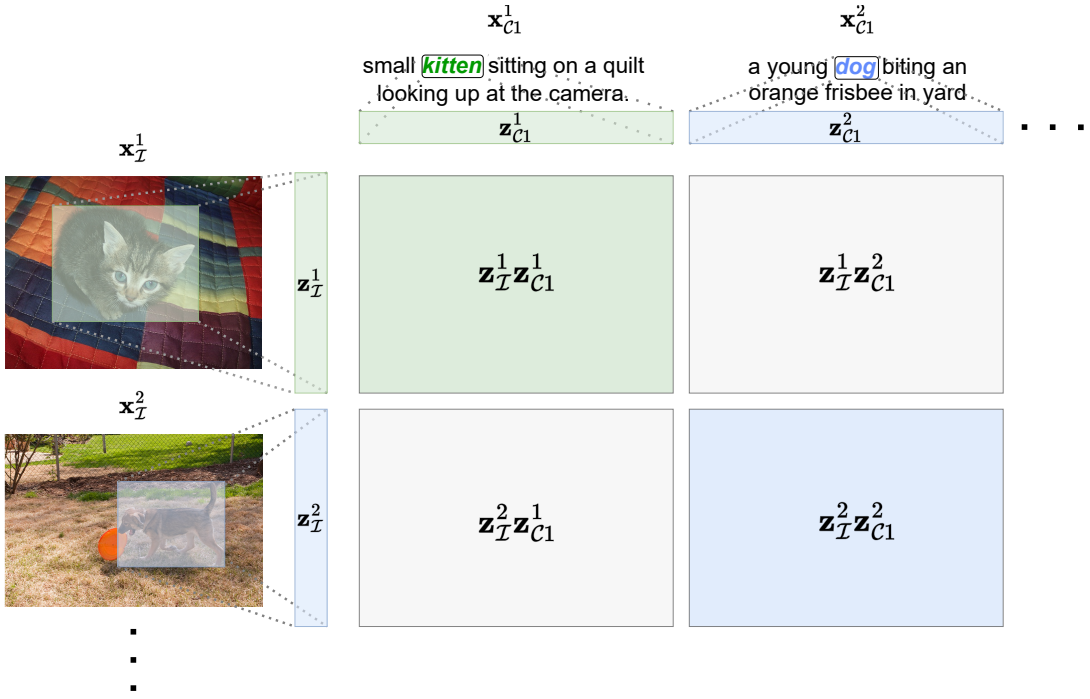


Figure 5.1: Visualization of *predictive feature suppression* using two examples from the MS-COCO Captions dataset. $x^*_{\mathcal{I}}$ and $x^*_{\mathcal{C}j}$ are input *images* and *captions*, respectively, and $z^*_{\mathcal{I}}$ and $z^*_{\mathcal{C}j}$ are the latent representations of the image and caption. We use the cosine similarity as similarity metric. The objective of a contrastive loss is to optimize the similarity scores on the diagonal of the similarity matrix, while minimizing the off-diagonal scores. In this small-scale training setup, if both the image and caption encoder only extract the concepts **kitten** and **dog**, the remaining concepts in both the images and captions are irrelevant to predicting the correct matching scores between the images and captions. The input features that are not needed to predict a match between an image and a caption are likely to be *suppressed* by the encoder. However, these features might be relevant during evaluation to predict a correct match.

rather than relying on an increased batch size to include difficult negative examples. The disadvantage of hard negative mining is that it can be computationally expensive (Chen et al., 2020b). Moreover, the MS-COCO dataset contains many visually similar images (Parekh et al., 2020); when a similar image is mined as a hard negative, it will be considered as a negative w.r.t. the query, which may create conflicting and incorrect supervision signals.

The autoencoding paradigm (Hinton and Salakhutdinov, 2006) provides an alternative solution to reduce predictive feature suppression by learning latent data representations that contain as much of the important input features as possible. Using the information bottle-neck principle, the encoder should

compress the input information into a low-dimensional representation while preserving as much as possible of the input features. Combining autoencoding with contrastive learning should prevent the image and caption encoder from learning features that are only needed to solve the contrastive optimization objective. Therefore, a logical step is to add a decoder to the learning algorithm that decodes the original input from either the caption or image representation (or both). However, adding a decoder on top of the image representations, as in (Li et al., 2020b), is sub-optimal for the ICR task. The captions provided for each image are already a dense summary of the image; reconstructing every pixel in the image results in image representations that contain too much local information, which is irrelevant for the ICR task. A more natural choice would be to decode the input caption rather than the image, but adding a decoder on top of the caption representations might not result in a reduction of predictive feature suppression. Strong textual decoders can reduce a reconstruction loss by mainly relying on the learned language model (Lu et al., 2021a). The input for this decoder (the latent caption representation) can mostly be ignored while correctly decoding the input sequence.

Our proposed solution. To address the disadvantages of current approaches to mitigating predictive feature suppression, viz. (i) high costs (in terms of compute and data), and (ii) reconstruction of the input caption and images in the input space, we introduce *latent target decoding* (LTD). For each caption in the training set, we generate a *latent target* representation by using a general-purpose sentence encoder. We train an image and caption encoder that can be trained in a resource-constrained setup, using a standard contrastive learning objective. Next to that, we add an extra decoder to the learning algorithm. We decode the information of the caption in a latent space of the sentence encoder. Thus, the decoder cannot rely on learning a dataset-specific language model to decode the input, and the caption representation learned by the caption encoder should contain all input features that are needed to decode the latent target. By reconstructing this latent target representation we aim to reduce predictive feature suppression by the caption encoder, which should result in representations that generalize better to the evaluation task. See Figure 5.2 in Section 5.3 for a high-level overview of our LTD method. LTD only requires an additional target representation for each caption and a simple feed-forward decoder network, and can be combined with any ICR method that uses a separate caption and image encoder to compute a global representation of the

input data. LTD does *not* depend on (i) additional training data, (ii) extra manual data annotation or (hard) negative mining, or (iii) significantly more computational resources. In this chapter, we focus on *resource-constrained* ICR methods that are trained from scratch on the Flickr30k or MS-COCO dataset on a single GPU.

If we were to add LTD to the learning algorithm, the overall training objective would become a multi-task loss: a contrastive and reconstruction loss. However, multi-task losses are difficult to optimize (Malkiel and Wolf, 2021). The reconstruction loss should serve as an extra regularizer rather than the main learning objective. We also do not want the caption encoder to mainly focus on the reconstruction objective, since that can harm the contrastive utility of the representations. Therefore, we implement LTD as an optimization constraint. In this manner, we can target a specific value for that loss function. The main training objective is to minimize the contrastive loss, given the constraint that the reconstruction loss is below a certain bound value. Similar to (Rezende and Viola, 2018; Rozendaal et al., 2020), we implement the reconstruction loss constraint using a Lagrange multiplier (Platt and Barr, 1987); the two losses are scaled automatically such that the reconstruction bound is met while minimizing the contrastive loss.

Our main findings. In this chapter, we contribute the following:

(i) The proposed constraint-based LTD reduces predictive feature suppression and improves the generalizability of learned representations, as it outperforms ICR baselines that are only optimized by using a contrastive loss. We measure the reduction of predictive feature suppression by using the standard evaluation metrics for the ICR task.

(ii) Implementing LTD as a dual loss, as opposed to an optimization constraint, does not reduce predictive feature suppression. Our analyses suggest that optimizing the reconstruction loss only until a specific bound value is met, results in better evaluation performance than minimizing the reconstruction loss as a dual loss.

(iii) LTD can be used in combination with different contrastive losses, for example, InfoNCE (van den Oord et al., 2018) and the triplet loss (Faghri et al., 2018), and it can be combined with a wide variety of ICR methods that can be optimized in a resource-constrained setup, such as VSRN (Li et al., 2019a) and TERN (Messina et al., 2020b).

Below, we first cover related work, then introduce the proposed LTD method, before presenting our experimental setup, discussing the outcomes of our experiments, and concluding.

5.2 RELATED WORK

5.2.1 Image-caption retrieval

Neural architectures for ICR. We focus on ICR methods that compute a global representation for both the image and caption. In general, an ICR method consists of two encoders: one to encode the image and one to encode the caption into a latent representation (Faghri et al., 2018; Li et al., 2019a; Chun et al., 2021; Jia et al., 2021). Most work on ICR focuses on new network architectures to learn multi-modal feature representations. State-of-the-art results have been obtained using graph neural networks (Li et al., 2019a; Liu et al., 2020; Wang et al., 2020; Diao et al., 2021) to represent visual relations in scenes as a graph, or attention mechanisms to align words in the caption with specific regions in the input image (Lee et al., 2018; Chen et al., 2019a; Wang et al., 2019; Chen et al., 2020a; Yu et al., 2021b; Zhang et al., 2022). Li et al. (2019a) combine a caption encoder-decoder with the image encoder to add extra training signals to the learning algorithm. These methods are only trained and evaluated on the Flickr30k and MS-COCO datasets. Recently, there has been a shift to transformer-based (Vaswani et al., 2017) network architectures for both the image and caption encoder. Messina et al. (2020a) and Messina et al. (2020b) introduce a transformer-based network architecture solely trained for the ICR task. Since then, several transformer-based methods have been introduced (Lu et al., 2019; Chen et al., 2020d; Li et al., 2020c; Jia et al., 2021; Li et al., 2021; Li et al., 2022a); some combine the image and caption encoder into one unified architecture. These methods are all (pre-)trained on a large amount of additional training data and most are not trained for the ICR task specifically, but as general-purpose vision-text models.

Hard negative mining. Few publications have looked into the improvement of contrastive optimization for ICR methods. Faghri et al. (2018) introduce a new formulation of the triplet loss that only considers the hardest negative candidate in the training batch instead of all negative candidates, which sig-

nificantly improved the evaluation scores on the ICR benchmarks. Since then, many ICR methods (Lee et al., 2018; Li et al., 2019a; Chen et al., 2020a; Liu et al., 2020; Messina et al., 2020a; Messina et al., 2020b; Wang et al., 2020; Diao et al., 2021; Yu et al., 2021b) have used this loss function for optimization. Chen et al. (2020b) introduce an offline hard negative mining approach for ICR in order to overcome the limitations of in-batch negative mining. Instead of mining an in-batch hard negative, they mine additional negative candidates, for each query, over the entire training set to learn from so-called harder-to-distinguish negatives.

One-to-many problem. Chun et al. (2021) focus on the one-to-many problem in ICR. An image can be described by many captions. However, most methods in ICR learn one representation for the image, which should match with a number of different captions. They propose a probabilistic ICR method, where images and captions are represented as probability distributions in a shared latent space instead of a point representation. Although their method does not focus on contrastive optimization, it addresses predictive feature suppression by learning a distribution over features instead of a point prediction of features. Chun et al. (2021) also propose the plausible match metric, a heuristic for identifying missing positive examples by finding images that contain similar objects (i.e., plausible matches) and considering these in the evaluation.

Biten et al. (2022) propose semantic adaptive margin (SAM). Instead of using the binary relevance annotation between images and captions (of the Flickr30k and MS-COCO datasets) for the triplet loss computation, the authors propose an adaptive margin to model the many-to-many relation between images and caption. The standard triplet loss uses a fixed margin parameter α. SAM dynamically assigns a unique distance value to the triplets in the training batch, based on the semantic similarity between an image and caption. In contrast, in this chapter, we do not change the formulation of the contrastive loss. We add an extra optimization objective to the learning algorithm to prevent predictive feature suppression.

Upshot

Unlike most previous work, we do not focus on the network architecture to improve the ICR performance. Similar to (Chun et al., 2021), we focus on small-scale learning set-ups to train an ICR method from scratch to show the strength of our method in a resource-constrained setting. Our proposed approach in-

corporates autoencoding into the learning algorithm in order to reconstruct the input caption to reduce predictive feature suppression.

5.2.2 Contrastive representation learning

Contrastive learning losses are used to learn discriminative representations of the input data that can be used to contrast positive and negative pairs of information in a latent space. These loss functions have made a big impact in representation learning, whether self-supervised (van den Oord et al., 2018; Chen et al., 2020c) or supervised (Karpukhin et al., 2020; Radford et al., 2021). Although ICR is a supervised contrastive learning task, some of the theoretical findings about self-supervised contrastive learning apply to supervised settings as well.

Self-supervised contrastive learning. A common approach to learn general-purpose representations in a self-supervised manner, is to create two (matching) views of the same (or similar) data point(s) by applying different augmentations (Chen et al., 2020c) or by splitting the data into parts (van den Oord et al., 2018) (i.e., predicting the future). The two positive views are contrasted with other negative samples in the training batch. The goal is to learn encoders that are invariant under these augmentations and that can discriminate between positive and negative pairs. How well self-supervised representations generalize to different settings, after training, is often assessed using a down-stream evaluation task, such as object classification (Chen et al., 2020c) or speaker identification (van den Oord et al., 2018).

Some work examines data augmentation to learn strong feature representations. Good augmentations retain task-relevant information while removing task-irrelevant nuisances (Tian et al., 2020b). The main purpose of removing task-irrelevant nuisances is to prevent encoders from using this information as predictive features during training. Xiao et al. (2021) show that the features needed to learn good representations depend on the down-stream task. ICR does not depend on augmentations to generate positive and negative pairs. These pairs are given by the annotations of the benchmark datasets (Lin et al., 2014; Young et al., 2014). The difficulty of the discrimination task (and hence the learned features) mainly depends on which candidates are present in the training batch.

The generalizability of contrastive learning methods is also influenced by the number of (hard) negatives present in a training batch. In general, the larger the number of in-batch negatives, the higher the down-stream evaluation performance (Chen et al., 2020c). Some work has focused on methods to increase the number of negatives during training (He et al., 2020) or on applying hard-negative mining strategies to increase the number of hard negatives in the batch (Lindgren et al., 2021; Xiong et al., 2021). Since we are focusing on a resource-constrained setup in this chapter, scaling up the batch size to increase the number of (hard) negatives is not a feasible solution. Moreover, the MS-COCO dataset contains many visually similar images (Parekh et al., 2020). Mining visually similar images as (hard) negatives will result in a suboptimal supervision signal, which makes hard-negative mining also not a feasible approach to reduce shortcut feature suppression.

Shortcut feature representations. Robinson et al. (2021) show that the contrastive InfoNCE loss (van den Oord et al., 2018) does not guarantee avoidance of shortcut feature representations. Shortcut feature representations are solutions that suppress predictive input features, i.e., a shortcut to discriminate between matching/non-matching candidates. The features learned by the InfoNCE loss depend on the difficulty of instance discrimination during training. If the instance discrimination task is easy to solve during training, the model will learn shortcut features. Especially in a resource-constrained ICR training setup, the contrastive objective is easy to solve, since there is only a limited number of (hard) negative samples in the training batch, which will result in shortcuts/predictive feature suppression.

Feature suppression among competing features. Chen et al. (2021a) introduce the notion of feature suppression among *competing features*. (Chen et al., 2020c) show that, for example, the SimCLR method (Chen et al., 2020c) when trained without the crop or color augmentation (which randomly crops or shifts the color distribution of an image), shows a significant drop in performance on the down-stream evaluation task. Apparently, the (color) pixel distribution of the image is a sufficient predictive feature to match two views of the same image during training. However, these features do not generalize well to a down-stream evaluation task, such as object classification. The desired predictive features of the input image (i.e., the object class and its properties) are suppressed by competing features (i.e., the color distribution of the image). Chen et al. (2021a) refer to this phenomenon as *feature suppression among competing*

features. Feature suppression among competing features is closely related to work by Hermann and Lampinen (2020), who show that in the presence of multiple redundantly predictive features, deep neural models prefer one of the features over the other, while the other feature is suppressed. Chen et al. (2021a) add artificially generated features (i.e., MNIST digits) as an extra overlay to images. They show that the "easy" predictive features (the MNIST digits) are preferred by a deep neural encoder model over the real predictive features (i.e., the object class) when optimizing with a contrastive learning loss. Chen et al. (2021a) conclude that contrastive losses rely on easy-to-detect features that solve the contrastive objective, while suppressing the remaining (partly irrelevant) information.

Predictive feature suppression is a prominent problem in resource-constrained contrastive ICR. Captions often describe multiple aspects of a scene. However, in a resource-constrained contrastive setup, only one (or a few) of the aspects that are described in the caption is likely to be sufficient to match with the positive candidate (i.e., the image) during training due to the limited number of negative candidates in the training batch. To mitigate this problem of predictive feature suppression for resource-constrained contrastive ICR, we need an extra optimization objective that is independent of the negative samples in the training batch.

Autoencoding. An approach to reduce predictive feature suppression is autoencoding (Hinton and Salakhutdinov, 2006). Autoencoding can be combined with a contrastive learning loss and reduces predictive feature suppression without depending on sampling (hard) negative candidates. To learn high-quality text sequence embeddings for the dense passage retrieval task, Lu et al. (2021a) add a weak decoder on top of a document encoder to reconstruct the original document. To make image encoders more robust against shortcut features, Li et al. (2020b) add a decoder on top of the image encoder to decode the input image.

Upshot

To reduce predictive feature suppression in a resource-constrained ICR task, we introduce *latent target decoding* (LTD). LTD reduces predictive feature suppression, without focusing on the difficulty of the contrastive discrimination task. LTD requires neither a large number of negative samples nor hard neg-

ative mining strategies. Unlike other methods that reconstruct the input data, we reconstruct the input information of the caption in a latent space instead of the input space.

5.3 METHOD

In Table 5.A.1 in Appendix 5.A, we provide an overview of the symbols and variables used throughout this chapter. We start with preliminaries and then discuss the InfoNCE contrastive loss and why autoencoding captions in the input space is not a solution to reduce predictive feature suppression. Finally, we introduce latent target decoding (LTD) to reduce predictive feature suppression for recourse-constrained ICR. In Figure 5.2 we provide an overview of LTD.

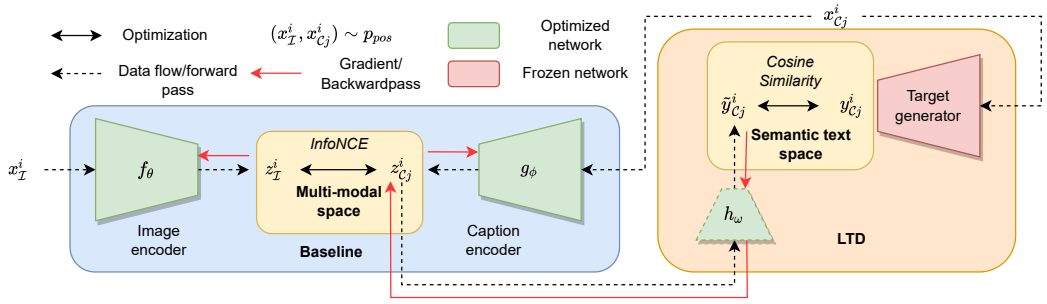


Figure 5.2: Overview of latent target decoding (LTD). The baseline (left) consists of a general image-caption retrieval framework with an image and caption encoder. The encoders are trained by using the contrastive InfoNCE loss. To reduce predictive feature suppression we add latent target decoding to the baseline ICR method (right). This extra decoder decodes the information of the input caption in a latent space of a general-purpose sentence encoder. The decoder is not used during inference, which we indicate by the dashed line around the model $h_\omega(\cdot)$.

5.3.1 *Preliminaries and notation*

Notation

We follow the notation introduced in previous work (Brown et al., 2020a; Chen et al., 2020c). For the ICR task we use a multi-modal dataset $\mathcal{D} =$

$\{(\boldsymbol{x}^i_{\mathcal{I}}, \boldsymbol{x}^i_{\mathcal{C}1}, \ldots, \boldsymbol{x}^i_{\mathcal{C}k}), \ldots\}_{i=1}^{N}$. This dataset consists of N image-caption tuples. Each tuple contains one image $\boldsymbol{x}^i_{\mathcal{I}}$ and k captions $\boldsymbol{x}^i_{\mathcal{C}j}$, where $1 \leq j \leq k$, that describe the scene depicted in the image. At each training iteration, we randomly sample a batch $\mathcal{B}$ of image-caption pairs from $\mathcal{D}$. Per image, one of the k captions is sampled per training iteration; together, this image and caption form a positive (or matching) image-caption pair. Each caption is used once during a training epoch.

The image and caption encoder are trained for two tasks: *image-to-text* (i2t) and *text-to-image* (t2i) retrieval. Thus, each image and caption in $\mathcal{B}$ is used as a query $\boldsymbol{q}$. We denote the matching candidate in the other modality as $\boldsymbol{v}^+$. All other candidates in $\mathcal{B}$, in the other modality, are considered as negative candidates $\boldsymbol{v}^-$. The set of all negative candidates for query $\boldsymbol{q}$ in batch $\mathcal{B}$ is $\mathcal{S}^-_{\boldsymbol{q}}$, where $\boldsymbol{v}^- \in \mathcal{S}^-_{\boldsymbol{q}}$.

Contrastive baseline model

The baseline (BL) ICR framework in this chapter consists of two encoders. The image encoder $f_\theta(\cdot)$ takes an image $\boldsymbol{x}^i_{\mathcal{I}}$ as input and encodes this image into a latent representation $\boldsymbol{z}^i_{\mathcal{I}} := f_\theta(\boldsymbol{x}^i_{\mathcal{I}})$. The caption encoder $g_\phi(\cdot)$ takes a caption as input and encodes this caption into a latent representation $\boldsymbol{z}^i_{\mathcal{C}j} := g_\phi(\boldsymbol{x}^i_{\mathcal{C}j})$. The vectors $\boldsymbol{z}^i_{\mathcal{C}j}$ and $\boldsymbol{z}^i_{\mathcal{I}}$ have the same dimensionality and are normalized on the unit sphere. The encoders are only optimized by minimizing a contrastive learning loss $\mathcal{L}_{con}$. Our goal is not to obtain the highest possible evaluation performance, but to show the strength of LTD on resource-constrained training setups.

5.3.2 Contrastive loss

To train the image and caption encoder, we use the InfoNCE contrastive loss (van den Oord et al., 2018; Chen et al., 2020c). The InfoNCE loss is a popular loss function for self-supervised representation learning (Chen et al., 2020c; He et al., 2020) and multi-modal representation learning (Jia et al., 2021; Radford et al., 2021; Yuan et al., 2021). To keep the notation simple and intuitive, we

use $\boldsymbol{q}$ and $\boldsymbol{v}$ for the latent representations computed by the caption and image encoder and not $\boldsymbol{z}_{\mathcal{C}_j}$ and $\boldsymbol{z}_{\mathcal{I}}$. The InfoNCE loss is defined as follows:

$$\mathcal{L}_{con} = \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{q},\boldsymbol{v}^+)\in\mathcal{B}} -\log \frac{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^-\in\mathcal{S}_{\boldsymbol{q}}^-}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)}. \tag{5.1}$$

$\mathcal{L}_{con}$ in Eq. 5.1 is minimized when, given a query $\boldsymbol{q}$, the cosine similarity score with the positive candidate $\boldsymbol{v}^+$ is high (i.e., ≈ 1), while the similarity scores with the negative candidates $\boldsymbol{v}^- \in \mathcal{S}_{\boldsymbol{q}}^-$ in the batch are as low as possible; τ serves as a temperature parameter. The main objective of a contrastive learning loss for the ICR task is to learn data representations that can be used to discriminate between similar and dissimilar image-caption pairs. However, there is no constraint that enforces the encoders to learn representations that contain all available input information to make this discrimination, which is what we add.

Gradient w.r.t. the input representations

To show some important properties of the InfoNCE loss, we provide the derivative of $-\mathcal{L}_{con}$ w.r.t. the input in Eq. 5.2 (Chen et al., 2020c):

$$Z(\boldsymbol{q},\boldsymbol{v}) = \frac{\exp(\boldsymbol{q}^T\boldsymbol{v}/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^-\in\mathcal{S}_{\boldsymbol{q}}^-}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)} \tag{5.2a}$$

$$\frac{\partial\mathcal{L}_{con}}{\partial\boldsymbol{q}}\tau = (1 - Z(\boldsymbol{q},\boldsymbol{v}^+))\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-\in\mathcal{S}_{\boldsymbol{q}}^-} Z(\boldsymbol{q},\boldsymbol{v}^-)\boldsymbol{v}^- \tag{5.2b}$$

$$\frac{\partial\mathcal{L}_{con}}{\partial\boldsymbol{v}^+}\tau = (1 - Z(\boldsymbol{q},\boldsymbol{v}^+))\boldsymbol{q} \tag{5.2c}$$

$$\frac{\partial\mathcal{L}_{con}}{\partial\boldsymbol{v}^-}\tau = -Z(\boldsymbol{q},\boldsymbol{v}^-)\boldsymbol{q}. \tag{5.2d}$$

$Z(\boldsymbol{q},\boldsymbol{v})$ returns the similarity score of candidate $\boldsymbol{v}$ w.r.t. the query $\boldsymbol{q}$, normalized by the sum of similarity scores of all candidates in the batch $\mathcal{B}$. The full derivations of the derivative $-\mathcal{L}_{con}$ are provided in Appendix 5.B. Based on Eq. 5.2, we infer the following properties:

(i) The update w.r.t. the query $\boldsymbol{q}$ (Eq. 5.2b), is a weighted sum over the positive candidate $\boldsymbol{v}^+$ and all negatives $\boldsymbol{v}^- \in \mathcal{S}_{\boldsymbol{q}}^-$. The query representation $\boldsymbol{q}$ will be pulled closer to $\boldsymbol{v}^+$, while being pushed away from all $\boldsymbol{v}^- \in \mathcal{S}_{\boldsymbol{q}}^-$. The weight value of each candidate, $Z(\boldsymbol{q},\boldsymbol{v})$ (Eq. 5.2a), depends on the similarity score with the query.

(ii) v^+ (Eq. 5.2c) will be pulled closer to the query representation (and the other way around).

(iii) All negatives v^- (Eq. 5.2d) will be pushed away from the query representation (and the other way around).

Without contrasting with negative candidates, the encoders will learn a trivial solution where latent representations collapse to a single point in the latent space (Jing et al., 2021). This means that the learned representation mainly depends on contrasting with negative candidates during training. If the representations v only contain a subset of the predictive input features (which still minimize the contrastive training objective), the query representation q (in the other modality) will be updated to match/mismatch these representations. The contrastive InfoNCE objective itself does not guarantee that all the predictive features in the input data are learned (Robinson et al., 2021) and mainly relies on easy-to-detect features to contrast between positive and negative pairs (Chen et al., 2021a).

Importantly, the query and candidate representations are in different modalities and therefore generated by different encoders. Hence, the update of the query and candidate representations is based on *fixed* representations in the other modality (e.g., the caption encoder can only try to match/not match with the representations of the image encoder and vice versa). By adding a constraint on the representations of one of the two modalities, the other modality encoder will follow automatically. Therefore, in order to prevent predictive feature suppression for the caption modality in a resource-constrained ICR setting, we add a constraint to the learning algorithm that forces the caption representation to be projected into the latent space of a general-purpose sentence encoder.

5.3.3 *Autoencoding reconstruction objective*

Autoencoding (Hinton and Salakhutdinov, 2006) is a natural choice for learning latent data representations that contain most of the important input features without relying on hard negative samples. To reconstruct the input caption from the encoded latent representation $z^i_{\mathcal{C}j}$, we introduce a decoder network $h_\omega(\cdot)$:

$$\widetilde{x}^i_{\mathcal{C}j} := h_\omega(z^i_{\mathcal{C}j}). \tag{5.3}$$

The decoder network $h_\omega(\cdot)$ takes the latent caption representation as input and outputs a reconstruction of the input caption $\widetilde{\boldsymbol{x}}^i_{\mathcal{C}j}$. To decode the input sequence from the latent representation, this latent representation should be a dense representation of the entire input sequence. The reconstruction loss, $\mathcal{L}_{rec}$, of a sequence of tokens, $x_i, \ldots, x_n$ of length n, is the negative log-likelihood of the input data:

$$\mathcal{L}_{rec} = -\sum_{t=1}^{n} \log\ p(x_t | x_{t-1:1}, \boldsymbol{z}^i_{\mathcal{C}j}). \tag{5.4}$$

Based on Eq. 5.4 it is clear that each predicted token x_t in the sequence is conditioned on: (i) the latent caption representation $\boldsymbol{z}^i_{\mathcal{C}j}$, and (ii) the already predicted sequence $x_{t-1:1}$.

As discussed in Section 5.1, a strong decoder will mainly rely on the learned language model and language patterns to decode the input sequence (Lu et al., 2021a). This implies that the input sequence can be decoded correctly while mainly ignoring $\boldsymbol{z}^i_{\mathcal{C}j}$, especially when t is large. Therefore, decoding the caption sequence in the input space is not guaranteed to reduce predictive feature suppression.

5.3.4 Latent target decoding

In Section 5.3.2 we argued why the contrastive InfoNCE loss is prone to predictive feature suppression, and in Section 5.3.3 we discussed why decoding a caption in the input space will not prevent predictive feature suppression. In this section, we introduce *latent target decoding* (LTD). LTD decodes the semantics of the input caption in the latent space of a general-purpose sentence encoder to reduce predictive feature suppression, which can be used in combination with a contrastive loss. LTD addresses the issues of decoding the caption in the input space. See Figure 5.2, at the beginning of Section 5.3, for a high-level overview of LTD for ICR.

For each caption $\boldsymbol{x}^i_{\mathcal{C}j}$ in the training dataset we generate $\boldsymbol{y}^i_{\mathcal{C}j}$, a *latent target representation*. The vector $\boldsymbol{y}^i_{\mathcal{C}j}$ is a dense vector representation. We assume that this vector contains all the (semantic) information of the caption, captured by a general-purpose language encoder. We use our decoding network h_ω to decode $\boldsymbol{y}^i_{\mathcal{C}j}$ instead of the input caption. By reconstructing a vector representation of the caption instead of the original input sequence, the reconstruction is not conditioned on the already predicted sequence of tokens. The latent target

decoder assumes conditional independence of each feature in the latent target. Therefore, the decoder cannot rely on conditional (language model) patterns in the data to reconstruct the input semantics. This implies that we force the decoder to rely completely on $\boldsymbol{z}^i_{\mathcal{C}j}$ to decode the latent target representation. LTD reduces predictive feature suppression by reconstructing the latent target from the caption embedding. To combine LTD with a contrastive loss, it is necessary to compute the similarity score between a *global* representation of the entire caption and the image embedding. ICR methods that compute the similarity score by using fragments of the caption and regions in the image cannot be combined with LTD as introduced in this chapter. If there is no global representation of the caption, it is not possible to enforce that all the semantic information from the target encoder will be distilled into the caption representation that is used for computing the similarity score.

Target decoding network

To decode $\boldsymbol{y}^i_{\mathcal{C}j}$ we use a three layer feed-forward decoder network:

$$h_\omega(\boldsymbol{z}^i_{\mathcal{C}j}) = \boldsymbol{W}^{(3)}\sigma\left(\boldsymbol{W}^{(2)}\sigma\left(\boldsymbol{W}^{(1)}\ \boldsymbol{z}^i_{\mathcal{C}j}\right)\right), \tag{5.5}$$

where σ is the ReLU non-linearity; h_ω takes the latent caption representation $\boldsymbol{z}^i_{\mathcal{C}j}$ as input and maps it to a reconstruction of the latent target representation $\widetilde{\boldsymbol{y}}^i_{\mathcal{C}j}$.

Loss function

To train h_ω, we use the cosine distance between $\widetilde{\boldsymbol{y}}^i_{\mathcal{C}j}$ and $\boldsymbol{y}^i_{\mathcal{C}j}$ as *reconstruction loss* $\mathcal{L}_{rec}$:

$$\mathcal{L}_{rec} = 1 - \frac{\widetilde{\boldsymbol{y}}^i_{\mathcal{C}j}}{\left\|\widetilde{\boldsymbol{y}}^i_{\mathcal{C}j}\right\|} \cdot \frac{\boldsymbol{y}^i_{\mathcal{C}j}}{\left\|\boldsymbol{y}^i_{\mathcal{C}j}\right\|}. \tag{5.6}$$

Minimizing the cosine distance is equivalent to minimizing the mean squared error of two vectors normalized on the unit sphere (Grill et al., 2020; Chen and He, 2021). By introducing an extra loss criterion, the overall training objective becomes a dual optimization problem. The *dual loss* $\mathcal{L}_{dual}$ is defined as follows:

$$\mathcal{L}_{dual} = \mathcal{L}_{con} + \beta\mathcal{L}_{rec}, \tag{5.7}$$

where β serves as a balancing parameter to scale the two losses.

Constraint-based optimization

By adding LTD to the learning framework, we introduce an extra loss component $\mathcal{L}_{rec}$. To effectively minimize $\mathcal{L}_{dual}$, we have to find the right value for the balancing parameter β in Eq. 5.7. This may require a considerable amount of manual tuning, and often one specific value for β does not generalize to different training settings. Besides that, $\mathcal{L}_{rec}$ is not the main training objective for the ICR tasks. The main reason we add $\mathcal{L}_{rec}$ to the learning algorithm is to reduce predictive feature suppression caused by solely optimizing the contrastive loss. We, therefore, argue that implementing LTD as an optimization *constraint* (Rezende and Viola, 2018; Rozendaal et al., 2020), as opposed to an optimization *objective*, might be more effective. Our goal, then, is to minimize the contrastive loss $\mathcal{L}_{con}$ given the constraint that the reconstruction loss is lower than or equal to a certain bound value η:

$$\min_{\theta,\psi,\omega} \mathcal{L}_{con} \text{ subject to } \mathcal{L}_{rec} \leq \eta. \tag{5.8}$$

We can implement this optimization constraint in combination with gradient descent by using the method of Lagrange multipliers:

$$\max_{\lambda} \min_{\theta,\psi,\omega} \mathcal{L}_{lag} = \mathcal{L}_{con} + \lambda \left(\frac{\mathcal{L}_{rec}}{\eta} - 1 \right). \tag{5.9}$$

The optimization objective is to minimize $\mathcal{L}_{lag}$ w.r.t. the model parameters θ, ψ, ω, while maximizing $\mathcal{L}_{rec}$ w.r.t. to the multiplier λ. The multiplier λ is tuned automatically by using stochastic gradient ascent with momentum. By optimizing λ with stochastic gradient ascent, the two losses will be balanced automatically during training such that the reconstruction constraint is met, while the contrastive loss is minimized by gradient descent.

Choice of latent target representation

To generate the latent target $\boldsymbol{y}_{\mathcal{C}j}$, we use a Sentence-BERT transformer model (Reimers and Gurevych, 2019; Song et al., 2020).[2] Sentence-BERT is a general-purpose sentence encoder that is trained on a large amount of data to capture the semantic input information. Thus, we expect these embeddings to be more general than those we learn for the resource-constrained contrastive ICR task, which makes them a suitable choice for the latent target representations $\boldsymbol{y}_{\mathcal{C}j}$.

2 https://huggingface.co/sentence-transformers/all-mpnet-base-v2

5.3.5 LTD vs. teacher-student framework

LTD is somewhat similar to a teacher-student framework used with knowledge distillation (Hinton et al., 2015). Indeed, the target generator can be seen as a teacher network and the caption encoder in combination with the target decoder as a student network. However, in contrast with knowledge distillation, the goal of LTD is *not* to closely mimic a teacher network. Instead, the goal is to learn caption representations that can be used for multi-modal contrastive-based retrieval while extracting as much of the textual semantic input information of the caption as possible.

5.4 EXPERIMENTAL SETUP

We design experiments aimed at showing: (i) a reduction of predictive feature suppression by using LTD, with a focus on the ICR task; (ii) the advantages of LTD over reconstructing the caption in the input space; (iii) the benefit of constraint-based optimization of LTD over dual loss optimization; and (iv) the generalizability of LTD to different contrastive losses and resource-constrained ICR methods that use different encoder network architectures. To facilitate reproducibility and further research of this chapter, we include the code with this chapter.[3]

5.4.1 Datasets

For training and evaluating our ICR methods, we use the two common ICR benchmark datasets: Flickr30k (Young et al., 2014) and MS-COCO (Lin et al., 2014). The Flickr30k dataset contains 31,000 image-caption tuples. We use the train, validate and test split from (Karpathy and Li, 2015), with 29,000 images for training, 1,000 for validation, and 1,000 for testing. MS-COCO consists of 123,287 image-caption tuples. We use the train, validate and test split from (Karpathy and Li, 2015); we do not use the 1k test setup. Both Flickr30k and MS-COCO come with $k = 5$ captions per image.

We also use the crisscrossed captions (CxC) dataset, which extends the MS-COCO validation and test set with additional annotations of similar cap-

3 https://github.com/MauritsBleeker/reducing-predictive-feature-suppression/

tions and images (Parekh et al., 2020), so as to evaluate whether LTD improves the evaluation scores by retrieving semantically similar candidates.

5.4.2 Implementation details

Unless otherwise specified, we use the following architectures for the target decoder, image encoder, and caption encoder. We use similar network architectures for the image and caption encoder as the ones used in (Chun et al., 2021), which are simple network architectures that can be trained using a limited amount of training data.

Image encoder. For the image encoder, we use a pre-trained ResNet-50 (He et al., 2016) network. We apply average pooling on the last convolutional layer followed by a projection head to map the image feature vector into a shared multi-modal latent space; the projection head has two feed-forward layers and a ReLU non-linearity.

Caption encoder. For the caption encoder, we use a bi-directional, single-layer, GRU network (Cho et al., 2014a). We use pre-trained GloVe embeddings (Pennington et al., 2014) as word embeddings. We use a similar projection head as for the image encoder (which does not share parameters) to map the caption embedding into the shared latent space.

Target decoding network. For the target decoding network, we use a three-layer feed-forward network as in Eq. 5.5. To generate the latent target representations, we use the HuggingFace *all-MiniLM-L6-v2* Sentence-BERT implementation. The target decoding network is trained together with the image and caption encoder. The target decoding network is *not* used during evaluation.

Input decoding network. We compare *latent* target decoding with *input* target decoding (ITD), which reconstructs the input caption in the input space (i.e., the input tokens). For ITD, we use a single-layer GRU (Cho et al., 2014a) decoder that reconstructs the input tokens in the caption (as explained in Section 5.3.3). We train the word embeddings for ITD from scratch. ITD is optimized with the negative log-likelihood loss (Eq. 5.4).

Training. Similar to (Chun et al., 2021), we use 30 warm-up and 30 fine-tune epochs, a batch size of 128, and a cosine annealing learning rate schedule with an initial learning rate of $2e^{-4}$. The Lagrange multiplier is initialized with a value of 1, bounded between 0 and 100, and is optimized by stochastic gradient

ascent with a fixed learning rate of $5e^{-3}$ and a momentum (to prevent λ from fluctuating too much) and dampening value of $\alpha = 0.9$. When we use $\mathcal{L}_{dual}$, we set β to 1. For the InfoNCE loss, we use a temperature value τ of 0.05. Evaluation scores of ICR methods tend to differ depending on the random seed used during training (Rao et al., 2022); to improve robustness, we apply stochastic weight averaging (SWA) (Izmailov et al., 2018); we take the average of 5 checkpoints, stored during the last 10% of the training iterations each epoch. For the reconstruction constraint bound η, we consider several values for all experiments, $\eta \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. When we apply ITD we use $\eta \in \{0.5, 1, 2, 3, 4, 5, 6\}$. All results are based on the best performing value of η.

Generalizability to different network architectures. LTD is a general method that can be combined with any global representation contrastive ICR method. To show that LTD works in combination with different network architectures, we apply LTD with multiple ICR methods that can be trained in a resource-constrained setup. To cover a wide spectrum of network architectures, we choose two methods that use different network architectures for the image and caption encoder.

VSRN. The visual semantic reasoning network (VSRN) (Li et al., 2019a) consists of an image and caption encoder that both compute a global representation of each input modality. The caption encoder consists of a single directed GRU, similar to the caption encoder used in (Faghri et al., 2018). The image encoder takes a set of pre-computed regions of interest as input generated by a ResNet-101 (He et al., 2016) backbone, pre-trained on visual genome (Krishna et al., 2016). This set of pre-computed visual features is considered a fully connected graph of regions in the input image. To perform reasoning on the graph of visual features, a multi-layer graph convolutional networks (Kipf and Welling, 2017) is used. Finally, to obtain one global representation of the entire image, a GRU is used to aggregate the regions of interest into one single representation. We use the same learning rate schedule and number of training epochs as in (Li et al., 2019a) and we use the model implementation as provided by the authors.[4] However, we modify the original VSRN model on two points:

(i) We use the same caption encoder as described in Section 5.4.2 instead of the *single* directed GRU used for the original VSRN model and use a

4 https://github.com/KunpengLi1994/VSRN

hidden dimensional of 1024 instead of 2048 for the caption and image representations. The goal of this chapter is not to show which specific network architectures perform best for the ICR task, but to show the generalizability of LTD in combination with different encoder networks.

(ii) The original VSRN model also comes with an additional caption decoder that decodes the input caption from the visual features. In this chapter, we investigate the reduction of predictive feature suppression for the general ICR framework, consisting of two encoders optimized by using a contrastive loss. If we would add LTD to the original VSRN method, we would have two reconstruction objectives and a contrastive loss. The main reason we use VSRN is for the use of graph convolution networks in the image encoder to show the generalizability of LTD in combination with different encoder networks. Therefore, we remove this caption decoder from the learning algorithm.

TERN. The transformer reasoning network (TERN) (Messina et al., 2020b) is a transformer-based ICR method that is solely trained and evaluated on the MS-COCO dataset. TERN consists of a pre-trained BERT (Devlin et al., 2019) caption encoder. The image encoder takes a set of pre-computed regions of interest as input (similar to VSRN) and consists of a stack of four transformer layers. The pre-computed features are only available for the MS-COCO dataset. Next, both the image and caption features are pushed through a stack of shared transformer layers. Although the weights of the last part of the caption and image encoder are shared, there is no (cross) attention between the two modalities; the representations of the images and captions are still computed independently. The image and caption CLS token is used as a global representation of both the image and the caption. We use the same learning rate schedule, dropout rate, number of training epochs, and data augmentations as in (Messina et al., 2020b) and use the model implementation as provided by the authors.[5]

In this chapter, we use ICR methods that are trained from scratch on the Flickr30k and MS-COCO dataset and that are trainable on a single GPU. That does not imply that some of the weights of our encoders are not initialized with pre-trained parameters. However, we only use pre-trained weights that are trained on a uni-modal task(s), and not for image-text matching specifically.

5 https://github.com/mesnico/TERN

5.4.3 Evaluation metrics

To measure the reduction of predictive feature suppression, we evaluate how well the learned encoders generalize to the ICR evaluation task. The more predictive features the encoders learn to capture, the better these encoders are able to retrieve the correct candidate given a query. The standard evaluation metric for ICR is the recall@k metric, with $k = \{1, 5, 10\}$. During training, we evaluate the model after each training epoch on the validation set. Similar to (Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a; Chen et al., 2020a; Chun et al., 2021), we select the model with the highest score on the validation set (using the sum of all recall@k scores as a metric) for evaluation on the test set.

Recall@k. For ICR, recall@k is implemented as the fraction of how many times a matching candidate is present in the top-k ranking (Karpathy and Li, 2015; Parekh et al., 2020). For the t2i task, there is only one matching image per query caption. For the i2t task, there are 5 matching captions per image. Only the highest-ranked caption per query is used for calculating the recall scores. When using the CxC annotations, for both i2t and t2i, we take the highest-ranked candidate.

R-precision. When extending the MS-COCO dataset with the CxC annotations, we have one or more matching candidates per query for both i2t and t2i. Like Chun et al. (2021), we also use r-precision (R-P) for evaluation; it measures the precision for a top-r ranking, where r is the number of matching candidates given a query.

nDCG. The standard evaluation metric for the ICR task, recall@k, mainly measures if the positive candidate is present in the top k of the ranking. However, this does not provide much insight into the overall quality of the ranking. To address the limitations of only using recall@k, (Messina et al., 2020b) start using nDCG as an additional evaluation metric for t2i retrieval. However, for t2i retrieval, there is only one positive image per query caption. To generate more positive images per caption query, images that have captions with a high overlap with the query caption, are also considered positive. As similarity measurements between the captions, ROUGE-L (Lin, 2004) and SPICE (Anderson et al., 2016) values are used. There are multiple re-annotations of MS-COCO available that provide multiple matching images per caption query; see, for example, (Parekh et al., 2020). However, these re-annotations are not used by Messina et al. (2020b) to compute the nDCG scores. To keep the evaluation con-

sistent, we use the same relevance labels as used in (Messina et al., 2020b). The nDCG relevance labels are only available for MS-COCO and not for Flickr30k.

5.5 RESULTS

In Section 5.5.1, we compare a contrastive ICR baseline with the same baseline combined with LTD. Next, in Section 5.5.2 we ask if similar results can be achieved with ITD as with LTD. In Section 5.5.3, we investigate the role of the optimization constraint and compare constraint-based LTD with LTD optimized as a dual loss. Finally, in Section 5.5.4, we ask whether LTD can be used in combination with a different contrastive loss function, and in Section 5.5.5 we show that LTD can be combined with a wide variety of resource-constrained ICR methods.

5.5.1 Contrastive ICR baseline vs. baseline + LTD

In Table 5.1 we compare the contrastive ICR baseline, which is optimized by using the contrastive loss $\mathcal{L}_{con}$ defined in Eq. 5.1, with the baseline combined with LTD. Based on Table 5.1, row 1.1, 1.4, 2.1.1, 2.4.1, and 2.4.2, 2.4.2, we observe that LTD optimized as a dual loss ($\mathcal{L}_{dual}$) with $\beta = 1$ does not convincingly (or only with a small margin) outperform the baseline ICR method, which is optimized solely in a contrastive manner, in terms of recall@k, nDCCG, and r-precision for both datasets and both i2t and t2i.

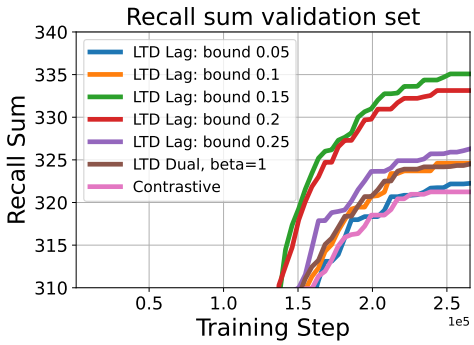
In contrast, when we implement LTD as an optimization constraint, by using $\mathcal{L}_{lag}$, row 1.5, 2.1.5, and 2.2.5, we observe that LTD consistently outperforms the baseline ICR methods on both Flickr30k and MS-COCO for both tasks (i2t and t2i) with a large margin. An increase in recall also comes with an increase in the r-precision scores and nDCG scores. Hence, features learned by constraint-based LTD perform better on the evaluation task, which is an indication of the reduction of predictive feature suppression.

Table 5.1: Recall@k, nDCG, and r-precision (R-P) evaluation scores for the Flickr30k and MS-COCO datasets (including the CxC annotations). We evaluate three loss functions $\mathcal{L}_{con}$, $\mathcal{L}_{dual}$ and $\mathcal{L}_{lag}$. We use three methods, the contrastive ICR baseline (BL), BL + input target decoding (ITD), and BL + latent target decoding (LTD). Boldface indicates the highest value for each evaluation metric per dataset. '-' indicates that it is not possible to compute the evaluation score for that dataset/experiment since the annotations are not available.

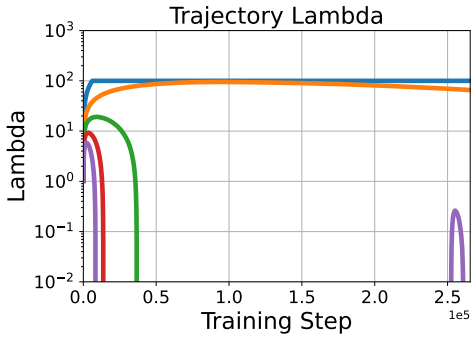
			i2t				t2i					
			R@k				R@k				nDCG	
#	Method	Loss	1	5	10	R-P	1	5	10	R-P	ROUGE-L	SPICE
						Flickr30k						
1.1	BL	$\mathcal{L}_{con}$	47.4	75.9	84.8	0.34	33.9	65.2	76.6	-	-	-
1.2	BL + ITD	$\mathcal{L}_{dual}$, β=1	45.7	74.0	84.4	0.33	33.7	65.1	75.8	-	-	-
1.3	BL + ITD	$\mathcal{L}_{lag}$, η=6	36.6	66.8	76.5	0.28	27.8	59.1	71.0	-	-	-
1.4	BL + LTD	$\mathcal{L}_{dual}$, β=1	46.1	75.3	84.1	0.34	34.0	65.9	77.4	-	-	-
1.5	BL + LTD	$\mathcal{L}_{lag}$, η=0.2	**49.6**	**78.7**	**86.4**	**0.37**	**36.7**	**68.4**	**79.3**	-	-	-
						MS-COCO						
2.1.1	BL	$\mathcal{L}_{con}$	33.7	64.4	76.6	0.24	24.2	53.5	67.0	-	0.6487	0.5729
2.2.1	BL + ITD	$\mathcal{L}_{dual}$, β=1	32.7	64.4	76.3	0.24	24.2	53.8	67.6	-	0.6496	0.5733
2.3.1	BL + ITD	$\mathcal{L}_{lag}$, η=4	28.4	59.2	72.2	0.22	22.0	50.4	64.5	-	0.6424	0.5638
2.4.1	BL + LTD	$\mathcal{L}_{dual}$, β=1	34.2	64.7	76.6	0.25	25.0	54.3	67.9	-	0.6510	0.5756
2.5.1	BL + LTD	$\mathcal{L}_{lag}$, η=0.15	**36.0**	**66.5**	**78.1**	**0.26**	**26.2**	**56.2**	**69.4**	-	**0.6531**	**0.5786**
						CxC						
2.1.2	BL	$\mathcal{L}_{con}$	36.1	68.1	80.2	0.22	26.7	57.6	71.0	0.23	-	-
2.2.2	BL + ITD	$\mathcal{L}_{dual}$, β=1	35.0	68.0	79.7	0.22	26.6	58.0	71.6	0.23	-	-
2.3.2	BL + ITD	$\mathcal{L}_{lag}$, η=4	31.0	62.9	75.8	0.20	24.6	54.8	68.9	0.21	-	-
2.4.2	BL + LTD	$\mathcal{L}_{dual}$, β=1	36.6	68.1	79.9	0.23	27.6	58.8,	72.0	0.24	-	-
2.5.2	BL + LTD	$\mathcal{L}_{lag}$, η=0.15	**38.4**	**70.4**	**81.5**	**0.24**	**28.9**	**60.4**	**73.3**	**0.25**	-	-

5.5.2 Latent target decoding vs. input target decoding

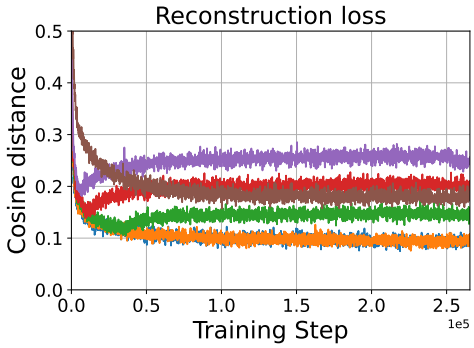
As argued in Section 5.3.3, decoding the caption in the input space will probably not result in a reduction of predictive feature suppression due to overfitting of the learned language model. To empirically show this, we also implemented a decoder that decodes tokens of the input caption (ITD) to reduce predictive



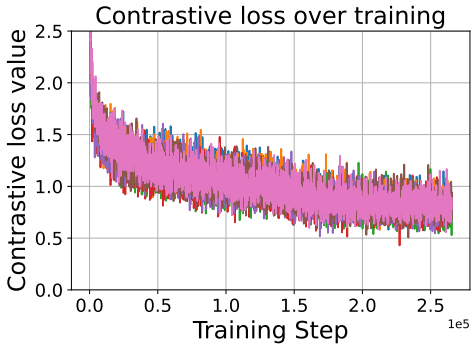
(a) Trajectory of the recall sum on the validation set during training.



(b) Trajectory of λ during training for different values of η. A log-scale is used for the y-axis.



(c) Trajectory of the reconstruction loss $\mathcal{L}_{rec}$.



(d) Trajectory of the contrastive loss $\mathcal{L}_{con}$.

Figure 5.3: Overview of constraint-based optimization on the evaluation metric and the optimization objectives. We train $\mathcal{L}_{lag}$ with four different values of $\eta \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$. All training steps are to the power of $1e5$.

feature suppression (see Section 5.4.2 for details). Based on row 1.2, 2.1.2, and 2.2.2 in Table 5.1, we conclude that implementing ITD as a dual loss does not result in improved recall@k scores, for most values of k, compared to the contrastive baseline. Surprisingly, when we implement ITD as an optimization constraint (with $\eta = 4$ for Flickr30k and $\eta = 6$ for MS-COCO, other values of η do not yield improvements) the evaluation scores are even lower (row 1.3, 2.1.3 and 2.2.3) than when implemented as a dual loss. We conclude that: (i) ITD does not reduce predictive feature suppression for ICR, and (ii) implementing ITD as an optimization constraint even hurts performance.

5.5.3 *The role of the optimization constraint*

What is the role of the optimization constraint when minimizing $\mathcal{L}_{rec}$ and what is the effect on the evaluation scores compared to using $\mathcal{L}_{dual}$? In Figure 5.3b we plot the trajectory of λ for different values of $\eta \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$ during training on the MS-COCO dataset. We also provide (i) the trajectory of the evaluation score (recall sum) over the validation set during training (Figure 5.3a), (ii) the trajectory of the reconstruction loss for different values of η and when optimized without using a constraint ($\mathcal{L}_{dual}$) (Figure 5.3c), and (iii) the trajectory of the contrastive loss for different values of η (Figure 5.3d). Based on Figure 5.3 we observe:

(i) λ increases until the optimization constraint is met (i.e., the bound η). The closer the reconstruction loss is to η, the slower the increase of λ. When the reconstruction constraint is met, λ decreases to 0 (Figure 5.3b).

(ii) λ is positive again when the reconstruction loss becomes higher than the bound η (Figure 5.3b, purple line).

(iii) The reconstruction loss converges to the value of η (Figure 5.3c). However, it is not possible to meet every value of η. E.g., $\eta = 0.05$ is too low to achieve for the model.

(iv) A lower reconstruction loss does not necessarily result in higher evaluation scores (Figure 5.3a). E.g., the recall sum is higher for $\eta = 0.15$ than for $\eta = 0.1$ or $\eta = 0.05$.

(v) The value and the development of the contrastive loss do not depend much on the value of the reconstruction loss (Figure 5.3d). E.g., a model optimized with $\mathcal{L}_{con}$ has the same contrastive loss trajectory as a model that is optimized with $\mathcal{L}_{lag}$ and $\mathcal{L}_{dual}$. Hence, the contrastive loss on its own does not provide a good indication of the performance on the evaluation task. Similar trajectories of the contrastive loss result in different evaluation scores (hence different learned representations).

When we implement LTD as a dual loss, there is always a gradient from the reconstruction loss w.r.t. the parameters of the caption encoder, until the reconstruction loss is 0. This is not the case when we implement LTD as a reconstruction constraint. When the constraint is met, λ drops to zero and there is no gradient anymore from the reconstruction loss. We can conclude that a constant gradient from the reconstruction loss does not improve the learned representations of the caption encoder in terms of evaluation scores.

The evaluation scores are higher when there is only a gradient until a certain reconstruction bound η is met.

5.5.4 *Generalizability w.r.t. contrastive loss*

In Section 5.3.2 we argued that the InfoNCE loss is prone to predictive feature suppression. A popular choice of contrastive loss function for ICR methods is the triplet loss with in-batch hard-negative mining (Faghri et al., 2018). The triplet loss with in-batch hard-negative mining is a special case of the InfoNCE loss, where the number of positives and negatives are each one (Khosla et al., 2020). Therefore, our line of reasoning in Section 5.3.2 holds for the triplet loss too.

To show the strength and generalizability of LTD to other contrastive losses, we run the same experiments as in Section 5.5.1 (only for LTD not for ITD), with the triplet loss instead of the InfoNCE loss as $\mathcal{L}_{con}$. To prevent the triplet loss from collapsing to the trivial solution, we added a batch normalization layer after the projection head, for both the image and caption encoder; we use a margin value of $\alpha = 0.2$ (Faghri et al., 2018; Li et al., 2019a; Messina et al., 2020b). Based on Table 5.1, we can observe that the highest recall@k scores also come with the highest r-precision and nDCG scores. Since the main goal of this experiment is to show that LTD can be used in combination with different contrastive losses we, therefore, only evaluate for recall@k.

Table 5.2 provides the recall@k scores for the Flickr30k and MS-COCO datasets. For both Flickr30k and MS-COCO the triplet loss with constraint-based LTD (see rows 3.1.3 and 3.2.3) results in higher evaluation scores than the InfoNCE loss with constraint-based LTD (see Table 5.1, rows 1.5 and 2.5). Our goal here is not to identify the best contrastive loss for ICR or LTD, but to show the generalizability of LTD to different contrastive losses. Moreover, using the triplet loss as $\mathcal{L}_{con}$ (see row 3.2.1), results in expected evaluation scores on the MS-COCO dataset (given the reproducibility work in (Bleeker and de Rijke, 2022)). Surprisingly, however, the evaluation scores for the Flickr30k dataset while using the triplet loss as $\mathcal{L}_{con}$ (see row 3.1.1) are lower than expected (when compared to Table 5.1, row 1.1). It is unclear why we observe these low evaluation scores for the Flickr30k dataset when only using the triplet loss as $\mathcal{L}_{con}$. In contrast, we observe that constraint-based LTD in combination with the triplet loss drastically improves the evaluation scores for the Flickr30k

dataset, which shows the strength of constraint-based LTD for improving ICR evaluation scores and also making the triplet loss more robust to predictive feature suppression and feature collapsing.

Table 5.2: Recall@k evaluation scores for the Flickr30k and MS-COCO datasets. For experiments 3.$*$, we use the triplet loss with in-batch hard-negative mining, as defined in (Faghri et al., 2018) instead of the InfoNCE loss (van den Oord et al., 2018) for $\mathcal{L}_{con}$. Boldface indicates the highest value for each evaluation metric per dataset.

			i2t			t2i		
#	Method	Loss	R@1	R@5	R@10	R@1	R@5	R@10
			Flickr30k					
3.1.1	BL	$\mathcal{L}_{con}$	12.8	33.2	45.4	11.1	32.6	46.6
3.1.2	BL + LTD	$\mathcal{L}_{dual}$, $\beta = 1$	17.1	42.7	56.4	13.1	40.5	55.7
3.1.3	BL + LTD	$\mathcal{L}_{lag}$, $\eta = 0.2$	**54.7**	**81.5**	**88.3**	**40.8**	**71.3**	**80.6**
			MS-COCO					
3.2.1	BL	$\mathcal{L}_{con}$	37.1	67.1	78.2	27.8	56.6	69.3
3.2.2	BL + LTD	$\mathcal{L}_{dual}$, $\beta = 1$	37.4	67.8	79.1	28.2	57.2	70.5
3.2.3	BL + LTD	$\mathcal{L}_{lag}$, $\eta = 0.2$	**39.1**	**69.3**	**80.6**	**29.6**	**59.4**	**72.2**

5.5.5 Generalizability w.r.t. network architectures

In this section, we consider whether LTD can be used in combination with different resource-constrained ICR methods that use different network architectures. To answer this question we use LTD in combination with the VSRN and TERN methods.

In line with the observations in Table 5.1, we observe in Table 5.3 that for both VSRN and TERN: (i) constraint-based LTD outperforms the contrastive baseline, and (ii) constraint-based LTD results in a stronger performance improvement than implementing LTD as a dual loss.

In line with the results in (Messina et al., 2020b), the VSRN baseline outperforms the TERN baseline in terms of Recall@k. However, the difference in nDCG scores between the two models is relatively small. Although constraint-based LTD outperforms both the baseline and LTD implemented as a dual

Table 5.3: Recall@k and nDCG evaluation scores for the Flickr30k and MS-COCO datasets using the VSRN and TERN network architectures. Boldface indicates the highest value for each evaluation metric per dataset. '-' indicates that it is not possible to compute the evaluation score for that dataset/experiment since the annotations are not available.

			i2t			t2i				
			R@k			R@k			nDCG	
#	Method	Loss	1	5	10	1	5	10	ROUGE-L	SPICE
			VSRN							
			Flickr30k							
5.1.1	BL	$\mathcal{L}_{con}$	60.3	85.6	90.8	44.9	74.0	83.3	-	-
5.1.2	BL + LTD	$\mathcal{L}_{dual}$, $\beta = 1$	59.6	86.6	91.6	44.3	74.7	83.5	-	-
5.1.3	BL + LTD	$\mathcal{L}_{lag}$, $\eta = 0.25$	**61.9**	**86.8**	**92.1**	**45.3**	**76.4**	**84.4**	-	-
			MS-COCO							
5.2.1	BL	$\mathcal{L}_{con}$	47.0	76.9	87.0	34.5	65.7	78.1	0.6779	0.6080
5.2.2	BL + LTD	$\mathcal{L}_{dual}$, $\beta = 1$	45.9	77.8	87.6	34.6	66.2	78.3	0.6791	0.6088
5.2.3	BL + LTD	$\mathcal{L}_{lag}$, $\eta = 0.15$	**47.6**	**79.0**	**87.8**	**35.0**	**66.7**	**78.9**	**0.6797**	**0.6112**
			TERN							
			MS-COCO							
5.3.1	BL	$\mathcal{L}_{con}$	41.2	72.6	83.6	31.0	61.9	74.7	0.6648	0.5926
5.3.2	BL + LTD	$\mathcal{L}_{dual}$, $\beta = 1$	42.3	74.3	84.4	31.4	62.7	75.4	0.6684	0.5993
5.3.3	BL + LTD	$\mathcal{L}_{lag}$, $\eta = 0.2$	**44.1**	**74.8**	**85.7**	**33.6**	**64.6**	**76.9**	**0.6727**	**0.6059**

loss, improvements gained by using LTD in combination with VSRN are less convincing than for TERN.

The most consistent improvement in evaluation scores is obtained by combining LTD with TERN, which is a fully transformer-based ICR method. This is a substantially different architecture from the one in Table 5.1 and VSRN, and such transformer network architectures are the most prominent network architectures these days for multi-modal tasks (Lu et al., 2019; Chen et al., 2020d; Radford et al., 2021; Yuan et al., 2021). When only a limited amount of training data is available and one wants to make use of (partly) pre-trained transformer

networks for multi-modal contrastive learning, constraint-based LTD can help to significantly improve the evaluation scores for ICR.

Furthermore, TERN makes use of a pre-trained BERT (Devlin et al., 2019) model as a caption encoder. BERT is a general-purpose text encoder pre-trained on text only. An open question is still why to train a caption encoder, while we use a (frozen) general-purpose sentence encoder to generate the latent targets for LTD; why not use the target decoder directly as a caption encoder? The results in Table 5.3 show that fine-tuning a general-purpose language encoder (i.e., BERT) with a contrastive loss as a caption encoder results in lower evaluation scores than fine-tuning the caption encoder in combination with constraint-based LTD. This shows that LTD helps to extract features (i.e., not suppressing these features) from the input data that are relevant for the ICR task, that are not captured by either the pre-trained BERT model or by only using the contrastive optimization objective. In Appendix 5.C, we provide three qualitative ranking results for i2t retrieval using TERN and samples from the MS-COCO test set. Based on the examples it is clear that a baseline TERN does not represent specific concepts (i.e., predictive features) that are needed to rank the correct captions on top of the ranking, while TERN optimized with constraint-based LTD does represent these predictive features.

5.6 DISCUSSION AND CONCLUSION

In this chapter, we presented latent target decoding, a novel approach to reduce predictive feature suppression for contrastive resource-constrained ICR methods. Instead of reconstructing the captions in the input space, LTD reduces predictive feature suppression by reconstructing the input caption in the latent space of a general-purpose sentence encoder. By reconstructing the input caption, it is more difficult for the image and caption encoder to suppress predictive features that are not needed to solve the contrastive optimization objective.

Main findings. Our results show that constraint-based LTD obtains higher evaluation scores than both a contrastive ICR baseline and LTD implemented as a dual loss. This implies that we are able to reduce predictive feature suppression (and hence improve evaluation performance) by using constraint-based LTD, which does not require additional image-text training data or hard-

negative mining strategies. Furthermore, we show that constraint-based LTD consistently results in a bigger improvement in evaluation scores than implementing LTD as a dual loss. These results suggest that, instead of simply minimizing both the contrastive and reconstruction loss, better evaluation scores can be obtained by only optimizing the reconstruction loss until a certain bound value η is met. Finally, we show that constraint-based LTD can be combined with different contrastive learning losses and a wide variety of resource-constrained ICR methods. This means that we answered the fourth research question of this thesis positively: we can reduce predictive feature suppression for resource-constrained contrastive image-text representation learning by using LTD.

Implications. The results of this chapter show that in a resource-constrained setup, the evaluation performance of contrastive ICR methods can be substantially improved by using constraint-based LTD, without relying on more training data or hard-negative mining strategies. We, therefore, argue that, in a resource-constrained setup, LTD should be part of the standard ICR framework to mitigate the problem of predictive feature suppression. Furthermore, we argue that when one uses an additional reconstruction objective to reduce predictive feature suppression, this objective should be considered to be implemented as an optimization constraint instead of a dual loss.

Limitations. In this chapter, we use a general-purpose sentence encoder to generate our latent target representation $\boldsymbol{y}_{Ck}$. However, we need to assume that this latent target representation contains the relevant information of the input caption. Furthermore, the availability of a general-purpose sentence encoder is not always guaranteed (e.g., when working with low-resource languages).

For the ICR task, the predictive features are the features needed to retrieve the positive item from a set of candidates. We, therefore, measure the reduction of predictive feature suppression by using the standard ranking evaluation metrics, such as recall@k, r-precision, and nDCG. However, we do not explicitly know which features are causing the observed improvement in the evaluation scores by using LTD.

Future work. We have several directions for future work. First, we plan to examine if the choice of different target generators will result in different ICR evaluation scores. Moreover, we also want to look into generating latent target representations without relying on a pre-trained sentence encoder.

Another promising direction for future work is analyses on the exact role of the optimization constraint. In Section 5.5.3 we examine the role of the optimization constraint when training the image and caption encoder. When the optimization constraint η is met, λ (i.e., the balancing parameter of the two losses) drops to zero and the reconstruction loss no longer provides a gradient (Figure 5.3b). Although λ is (close to) zero for the majority of the training after the constraint is met, the evaluation scores on the validation set remain higher than when optimizing with $\mathcal{L}_{dual}$, with $\beta = 1$ (Figure 5.3a). This suggests that a constant gradient from the reconstruction loss does not benefit the training process, which is the case if LTD is implemented as a dual loss. We plan future research into the role of the optimization constraint, by trying constraint-based optimization for other multi-task optimization problems.

In this chapter, we focus on the reduction of predictive feature suppression for resource-constrained ICR methods. In Section 5.1 we argued that predictive feature suppression is less of an issue for models that are trained with large batch sizes since more information is needed to match the query with the positive candidate (due to a large number of negative candidates). However, it remains a promising direction for further research to investigate if and how constraint-based LTD can be used for either large-scale contrastive image-text representation learning or for fine-tuning. Prominent large-scale image-text matching methods, such as ALIGN (Jia et al., 2021), use noisy image-text pairs scraped from the internet. It is unclear if the target generator will provide useful targets (and hence a training signal) when the caption has a weak relation with its matching image (which is possible for noisy image-text pairs). It might be the case that the target generator mainly provides useful supervision signals when using high-quality human-curated datasets, such as Flickr30k and MS-COCO.

Finally, we suggest working on methods to measure which features are responsible for the gained improvement in evaluation performance. A logical choice would be to use feature attribution methods. However, different feature attribution methods tend to disagree with each other, for both RNN and transformer-based models (Neely et al., 2021). Therefore, the choice of feature attribution method will influence the analyses of which predictive features are better captured by using LTD. To gain further insights into which features are captured by the learned encoders, we recommend developing task-specific feature attribution methods that can measure the reduction of predictive

feature suppression directly.

In the final research chapter of this thesis, we continue with the focus on predictive feature suppression for contrastive image-text representation learning. However, rather than framing the problem as predictive feature suppression, we redefine it as shortcut learning. We propose a framework that enables to inject synthetic shortcuts into image-text data, such that it becomes possible to measure to what extent contrastive image-text methods rely on shortcut solutions. Moreover, we examine existing shortcut learning reduction methods (e.g., LTD) using our proposed framework to investigate to what extent these methods reduce shortcut learning.

Chapter Appendix

5.A NOTATION AND VARIABLES

Table 5.A.1: Overview of the notation and variables used throughout Chapter 5.

Symbol	Explanation
$\mathcal{L}_{con}$	Contrastive loss. In this chapter, we either use the InfoNCE (van den Oord et al., 2018) or a triplet loss with in-batch hard-negative mining (Faghri et al., 2018).
$\mathcal{L}_{rec}$	Reconstruction loss of the input caption (i.e., *decoding loss*). In this chapter, we use the negative cosine similarity when using embeddings (*latent target decoding*) or the log-likelihood when we reconstruct the tokens of the input captions (*input target decoding*).
$\mathcal{L}_{dual}$	The sum of the contrastive loss and the reconstruction loss (i.e., the dual loss). The reconstruction loss is scaled by β.
$\mathcal{L}_{lag}$	The sum of the contrastive loss and the reconstruction loss, where the reconstruction loss is implemented as a Lagrange multiplier optimization constraint.
$\boldsymbol{q}$	Vector representation of a query, either an image or a caption.
$\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-$	Vector representation of a candidate. Given a query $\boldsymbol{q}$, a candidate is either matching ($\boldsymbol{v}^+$) or not matching ($\boldsymbol{v}^-$). Candidates are either images or captions.
$\mathcal{D}$	Dataset consisting of N image-caption tuples; each image $i \in N$ comes with k captions.
$\boldsymbol{x}^i_{\mathcal{I}}$	Input image i.
$\boldsymbol{x}^i_{\mathcal{C}_j}$	Input caption j that describes image i.
$\boldsymbol{z}^i_{\mathcal{I}}$	Latent representation of image i.
$\boldsymbol{z}^i_{\mathcal{C}_j}$	Latent representation of caption j that describes image i.

Continued on next page

Table 5.A.1 – continued from previous page

Symbol	Explanation
η	Reconstruction bound (or threshold). The reconstruction loss is only minimized up to the value of η.
λ	The Lagrange multiplier.
β	Balancing parameter to balance (or scale) two losses when using the dual loss.
$\mathcal{B}$	Batch with training samples.
$\mathcal{S}_q^-$	The set of all negative candidates $\boldsymbol{v}^-$, in a training batch, given query $\boldsymbol{q}$.
τ	Temperature parameter to scale the logist (i.e., cosine similarity) for the InfoNCE loss.
α	Margin parameter for the triplet loss.
$\boldsymbol{y}_{\mathcal{C}j}^i$	Latent target representation (i.e., semantic embedding produced by a SentenceBERT/language encoder) for caption j that describes image i.
$\widetilde{\boldsymbol{y}}_{\mathcal{C}j}^i$	Reconstructed latent target representation by the decoder network (i.e. LTD), for caption j that describes image i.
$\widetilde{\boldsymbol{x}}_{\mathcal{C}j}^i$	Reconstruction of the input tokens (i.e. ITD) by the decoder network for caption j that describes image i.
$f_\theta(\cdot)$	Image encoder parameterized by θ. Takes as input $\boldsymbol{x}_{\mathcal{I}}^i$. Outputs a latent (global) image representation $\boldsymbol{z}_{\mathcal{I}}^i$.
$g_\phi(\cdot)$	Caption encoder parameterized by ϕ. Takes as input $\boldsymbol{x}_{\mathcal{C}j}^i$. Outputs a (global) latent caption representation $\boldsymbol{z}_{\mathcal{C}k}^i$.
$h_\omega(\cdot)$	Decoder network parameterized by ω. Takes as input $\boldsymbol{z}_{\mathcal{C}j}^i$. Outputs a reconstuction of the input caption, either $\widetilde{\boldsymbol{y}}_{\mathcal{C}j}^i$ (LTD) or $\widetilde{\boldsymbol{x}}_{\mathcal{C}j}^i$ (ITD).

5.B GRADIENT OF THE INFONCE LOSS W.R.T. THE QUERY AND CANDIDATES

We start with the definition of the InfoNCE loss (van den Oord et al., 2018) using the notation introduced in Section 5.3, for one query candidate pair $(\boldsymbol{q}, \boldsymbol{v}^+)$ and a set of negative candidates $(\mathcal{S}_q)$:

$$\mathcal{L}_{con} = -\log \frac{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^-\in\mathcal{S}_q^-}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)} \tag{5.10a}$$

$$= -\left(\boldsymbol{q}^T\boldsymbol{v}^+/\tau - \log\left(\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)\right)\right) \tag{5.10b}$$

$$-\mathcal{L}_{con} = \left(\boldsymbol{q}^T\boldsymbol{v}^+/\tau - \log\left(\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)\right)\right). \tag{5.10c}$$

Next, we take the derivative of $-\mathcal{L}_{con}$ w.r.t. $\boldsymbol{q}$ (as also provided in (Chen et al., 2020c)):

$$-\frac{\partial\mathcal{L}_{con}}{\partial\boldsymbol{q}} = \boldsymbol{v}^+/\tau - \left(\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)\right)^{-1} \cdot \tag{5.11a}$$

$$\left(\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau)\boldsymbol{v}^+/\tau - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)\boldsymbol{v}^-/\tau\right)$$

$$= \boldsymbol{v}^+/\tau - \left(\frac{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)}\right)\boldsymbol{v}^+/\tau - \tag{5.11b}$$

$$\sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\left(\frac{\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) - \sum_{\boldsymbol{v}^-\in\mathcal{S}_q}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)}\right)\boldsymbol{v}^-/\tau.$$

Now let us define $Z(\boldsymbol{q}, \boldsymbol{v})$ (similar to Eq. 5.2a in Section 5.3):

$$Z(\boldsymbol{q}, \boldsymbol{v}) = \frac{\exp(\boldsymbol{q}^T\boldsymbol{v}/\tau)}{\exp(\boldsymbol{q}^T\boldsymbol{v}^+/\tau) + \sum_{\boldsymbol{v}^-\in\mathcal{S}_q^-}\exp(\boldsymbol{q}^T\boldsymbol{v}^-/\tau)}. \tag{5.12}$$

Next, we plug-in $Z(\boldsymbol{q}, \boldsymbol{v})$ into Eq. 5.11:

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{q}} = \boldsymbol{v}^+/\tau - Z(\boldsymbol{q}, \boldsymbol{v}^+)\boldsymbol{v}^+/\tau - \sum_{\boldsymbol{v}^- \in \mathcal{S}_q} Z(\boldsymbol{q}, \boldsymbol{v}^-)\boldsymbol{v}^-/\tau \tag{5.13a}$$

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{q}}\tau = \boldsymbol{v}^+ - Z(\boldsymbol{q}, \boldsymbol{v}^+)\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^- \in \mathcal{S}_q} Z(\boldsymbol{q}, \boldsymbol{v}^-)\boldsymbol{v}^- \tag{5.13b}$$

$$= \left(1 - Z(\boldsymbol{q}, \boldsymbol{v}^+)\right)\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^- \in \mathcal{S}_q} Z(\boldsymbol{q}, \boldsymbol{v}^-)\boldsymbol{v}^-. \tag{5.13c}$$

In a similar way, we can take the derivative of $-\mathcal{L}_{con}$ w.r.t. $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$:

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{v}^+} = \boldsymbol{q}/\tau - Z(\boldsymbol{q}, \boldsymbol{v}^+)\boldsymbol{q}/\tau \tag{5.14a}$$

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{v}^+}\tau = \left(1 - Z(\boldsymbol{q}, \boldsymbol{v}^+)\right)\boldsymbol{q}. \tag{5.14b}$$

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{v}^-} = -Z(\boldsymbol{q}, \boldsymbol{v}^-)\boldsymbol{q}/\tau \tag{5.15a}$$

$$-\frac{\partial \mathcal{L}_{con}}{\partial \boldsymbol{v}^+}\tau = -Z(\boldsymbol{q}, \boldsymbol{v}^-)\boldsymbol{q}. \tag{5.15b}$$

5.C RANKING EXAMPLES

In Figure 5.C.1 we provide three query images from the MS-COCO test set and the top 5 retrieved captions by TERN. We compare the TERN baseline (BL) and TERN optimized in combination with constraint-based LTD (BL + LTD). We selected three examples with a large difference in precision@5 between the BL and BL + LTD.

For all three examples, it is clear that the baseline ICR methods miss a concept (i.e., *predictive feature(s)*) that is needed to rank the ground-truth captions in the top 5. In the left example, the best matching captions according to the BL ignore that the man in the image *takes a photo*. In the middle example, the best matching captions, according to the BL, do not match on the fact that there is a *cup/mug*. In the right example, the best matching captions do not contain the concept of *teddy bears*. Clearly, an ICR method optimized in combination with LTD is able to match images and queries based on more fine-grained features in the images and captions than a baseline ICR method.

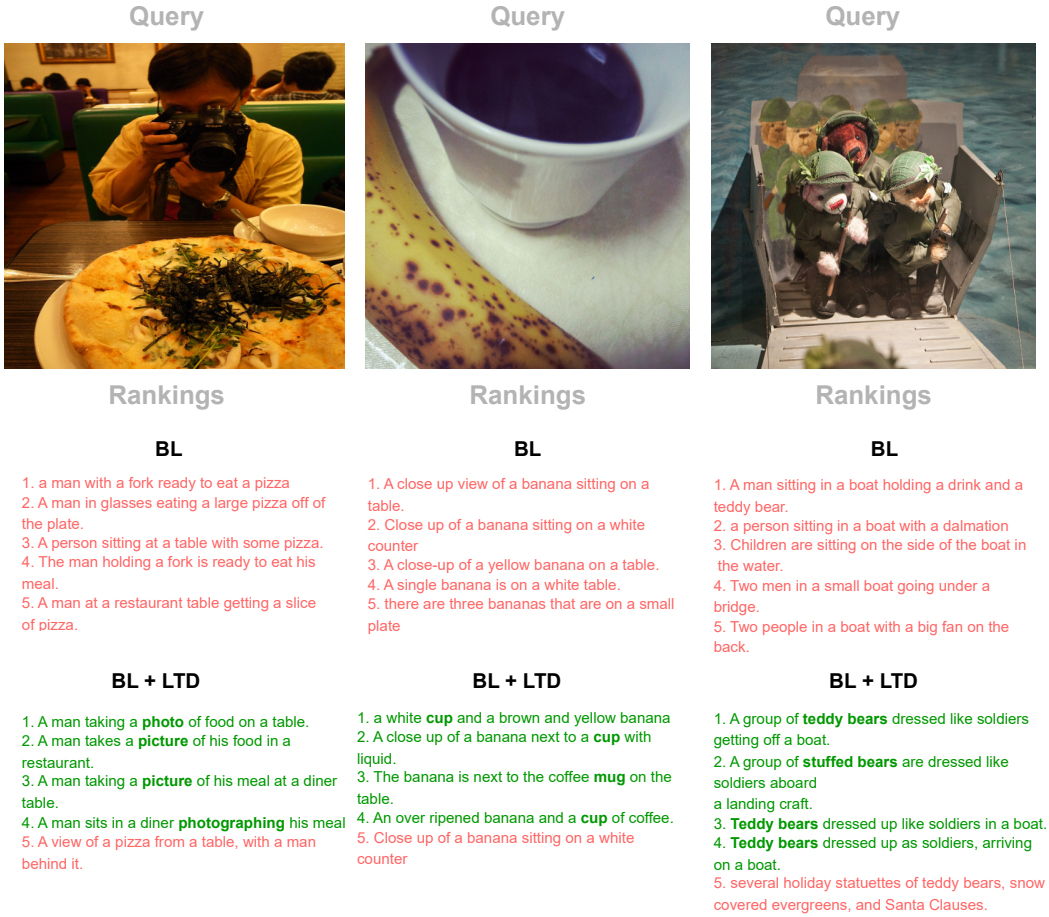


Figure 5.C.1: Three query images from the MS-COCO test set. For each image query we show the top 5 retrieved captions by TERN. We compare the TERN baseline (BL) and TERN optimized in combination with constraint-based LTD (BL + LTD). Ground-truth captions (i.e., matching) are indicated in green. Captions in red indicate captions that do not match with the query image.

6

Demonstrating and Reducing Shortcuts

In Chapter 5, we presented latent target decoding (LTD), a method to reduce predictive feature suppression for resource-constrained image-caption retrieval (ICR) methods. In this chapter, we continue our investigation on image-text representation, however, we refer to the task with a more general term: vision-language representation learning. One can argue that, when the representations learned by a contrastive loss do not contain certain predictive features in the input data, the model relies on a *shortcut* when minimizing the contrastive objective (Robinson et al., 2021) (i.e., the contrastive loss does not guide the learning process such that all relevant information in the input is extracted). Although we have shown in Chapter 5 that we can improve the generalizability of the learned representations by using LTD, the extent to which contrastive image-text methods rely on shortcuts during contrastive loss minimization remains an open question. In this chapter,[1] we try to answer this question by raising the fifth research question of this thesis:

Research Question 5: *Can we demonstrate and reduce shortcuts in contrastive image-text representation learning?*

To answer this question, we introduce a *synthetic shortcuts for vision-language (SVL)* framework. SVL is a training and evaluation framework that allows injecting synthetic shortcuts into image-text data. We show that contrastive image-text methods that are either trained from scratch or fine-tuned with data containing these synthetic shortcuts, predominantly learn features that represent the shortcut. Hence, we conclude that the contrastive loss is not sufficient to learn to extract all task-relevant (i.e., predictive) information in the image-text data. As a next step, we examine two methods to reduce shortcut

1 This chapter is based on (Bleeker et al., 2024).

learning in our SVL framework: (i) latent target decoding, and (ii) implicit feature modification. We find that both methods improve performance on the ICR evaluation task in some settings, however, they only partially reduce shortcut learning when training and evaluating with our SVL framework.

6.1 INTRODUCTION

Recent work on understanding the internal mechanisms of representation learning has brought to attention the problem of shortcut learning (Chen et al., 2021a; Robinson et al., 2021; Scimeca et al., 2022). While there are multiple definitions of shortcut learning (e.g., Geirhos et al., 2020; Wiles et al., 2022), in this chapter we define *shortcuts* as *easy-to-learn discriminatory features that minimize the (contrastive) optimization objective but are not necessarily sufficient for solving the evaluation task*. More specifically, we focus on the problem of shortcut learning in the relatively unexplored context of vision-language (VL) representation learning with multiple matching captions per image.

Contrastive learning (CL) plays a crucial role in VL representation learning. Despite the success of non-contrastive approaches, e.g., (Bardes et al., 2022), the dominant paradigm in VL representation learning revolves around either fully contrastive strategies (Faghri et al., 2018; Li et al., 2019a; Jia et al., 2021; Radford et al., 2021) or a combination of contrastive methods with additional objectives (Li et al., 2021; Li et al., 2022a; Zeng et al., 2022; Li et al., 2023a). It is standard practice in contrastive VL representation learning to sample batches of image-caption pairs and maximize the alignment between the representations of the matching images and captions (Radford et al., 2019; Jia et al., 2021). Given that the typical VL benchmarks, e.g., Flickr30k (Young et al., 2014) and MS-COCO Captions (Lin et al., 2014; Chen et al., 2015), are constructed in such a way that each image is associated with multiple captions, each caption can be seen as a different *view* of the image it describes. Therefore, CL with multiple captions per image can be seen as CL with multiple views, where each caption provides a different view of the scene depicted in the image.

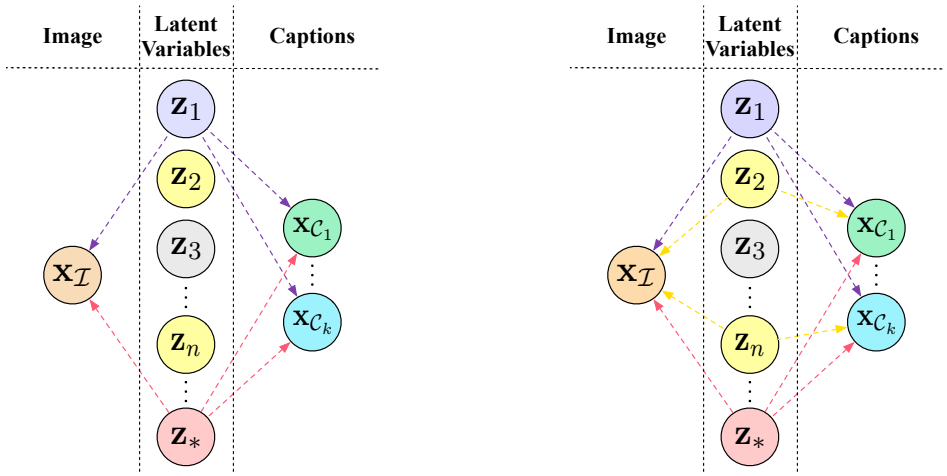
CL with multiple views, where each view represents a different observation of the same datapoint, has proven to be effective for general-purpose representation learning (Hjelm et al., 2019; Chen et al., 2020c; Tian et al., 2020a). The goal of multi-view (contrastive) representation learning methods is to learn

representations that remain invariant to a shift of view, which is achieved by maximizing the alignment between embeddings of similar views. A core assumption within the multi-view representation learning literature is that task-relevant information is shared across views whereas task-irrelevant information is not shared, given a downstream evaluation task (Zhao et al., 2017; Federici et al., 2020; Tian et al., 2020a; Shwartz-Ziv and LeCun, 2023).

An open challenge in the multi-view representation learning domain concerns *learning representations that contain task-relevant information that is not shared among different views, i.e., that may be unique for some views* (Shwartz-Ziv and LeCun, 2023; Zong et al., 2023). In the case of image-caption datasets where each image is paired with at least one corresponding caption, the captions matching the same image do not necessarily share the same information as each caption is distinct and may describe different aspects of the image (Biten et al., 2022). Furthermore, given the typical quality of captions of image-caption datasets (Chen et al., 2015), we assume that all information present in the captions is relevant. Hence, each image-caption pair may contain both *shared* task-relevant information, i.e., information shared across all the captions in the tuple, and *unique* task-relevant information, i.e., information not shared with other captions. Therefore, learning task-optimal representations for the image implies learning all task-relevant information that comprises both shared and caption-specific information.

Another problem of CL approaches is related to *feature suppression*. Shwartz-Ziv and LeCun (2023) argue that although contrastive loss functions lack explicit information-theoretical constraints aimed at suppressing non-shared information among views, the learning algorithm benefits from simplifying representations by suppressing features from the input data that are not relevant for minimizing the contrastive loss. Furthermore, Robinson et al. (2021) demonstrate that contrastive loss functions are susceptible to solutions that suppress features from the input data. In the case of VL, CL with multiple captions per image where at least one caption contains caption-specific information, the image representation can never have a perfect alignment with all matching captions. This is due to the misalignment that happens when encoding unique information for the other captions. Therefore, it is unclear whether contrastive methods are able to learn task-optimal representations, i.e., representations that contain all information present in the captions associated with the image, or if they learn only the minimal shared information, i.e., informa-

tion shared between the image and all captions that are sufficient to minimize the contrastive discrimination objective. An illustration of minimal shared information and a task-optimal representation is given in Figure 6.1.



(a) Minimal shared latent information.

(b) Task-optimal representation.

Figure 6.1: Synthetic shortcuts in the context of minimal shared latent information and task-optimal representation for vision-language representation learning with multiple captions per image. The purple color indicates a feature shared among the image and all captions (minimal shared information). The yellow color indicates caption-specific features (unique information). The grey color indicates features that are not present in both the image and any of the captions (task-irrelevant information). The red color indicates synthetic shortcuts. In this figure, we show the discrepancy between learning representations for the images and the matching captions with a strong alignment (i.e., high similarity) and learning representations that contain both the shared and caption-specific information (i.e., task-optimal).

Motivated by the abovementioned problems, we address the following question:

> *In the context of VL representation learning with multiple captions per image, to what extent does the presence of a shortcut hinder learning task-optimal representations?*

To answer this question, we investigate the problem of shortcut learning for VL representation learning with multiple captions per image. We do this by introducing the *synthetic shortcuts for vision-language* (SVL) framework for adding additional, easily identifiable information to image-caption tuples. The information that we add is represented as identifiers that are applied to both image

and caption; these identifiers do not bear any semantic meaning. The identifiers provide additional shared information between the image and captions, which is a subset of the total shared information between the image and the caption. Some examples are shown in Figure 6.D.1. The synthetic shortcuts framework allows us to investigate how much the encoder model relies on the added shortcut during training and evaluation, and hence how much of the relevant information is still captured if a shortcut solution is available. Overall, our SVL framework allows us to investigate the shortcut learning problem in a controlled way. We focus on image-caption retrieval (ICR) as an evaluation task because contrastive losses directly optimize for the ICR evaluation task, which assesses the quality of the learned representations by computing a similarity score between images and captions (Radford et al., 2021; Yuksekgonul et al., 2023). To investigate the problem, we run experiments on two distinct models: (i) CLIP (Radford et al., 2019), a large-scale model that we fine-tune; and (ii) VSE++ (Faghri et al., 2018), a relatively small model that we train from scratch. We evaluate the models' performance on the Flickr30k (Young et al., 2014) and MS-COCO (Lin et al., 2014; Chen et al., 2015) and benchmarks. The benchmarks are constructed in such a way that each image is associated with five captions and each caption represents a concise summary of the corresponding image.

Therefore, the contributions of this chapter are two-fold:

I **A framework for investigating the problem of shortcut learning for contrastive vision-language representation learning in a controlled way**: We introduce the *synthetic shortcuts for vision-language* framework. The framework enables the injection of synthetic shortcuts into image-caption tuples in the training dataset. We use the framework to investigate and understand the extent to which contrastive VL models rely on shortcuts when a shortcut solution is available. We run our experiments using CLIP and VSE++, two distinct vision-language models (VLMs). We evaluate the models' performance on the Flickr30k and MS-COCO benchmarks. We evaluate the effectiveness of contrastive VL models by comparing their performance with and without synthetic shortcuts. We demonstrate that both models trained from scratch and fine-tuned, large-scale pre-trained foundation models mainly rely on shortcut features and do not learn task-optimal representations. Consequently, we show that contrastive losses mainly capture the easy-to-learn discrimina-

tory features that are shared among the image and all matching captions, while suppressing other task-relevant information. Hence, we argue that contrastive losses are not sufficient to learn task-optimal representations for VL representation learning.

II **We present two shortcut learning reduction methods on our proposed training and evaluation framework:** We investigate latent target decoding (LTD) and implicit feature modification (IFM) using our SVL training and evaluation framework. While both methods improve performance on the evaluation task, our framework poses challenges that existing shortcut reduction techniques can only partially address, as the performance is not on par with models trained without synthetic shortcuts. These findings underline the importance and complexity of our framework in studying and evaluating shortcut learning within the context of contrastive VL representation learning.

6.2 BACKGROUND AND ANALYSIS

In this section, we present the notation, setup, and assumptions on which we base this chapter. Additionally, we conduct an analysis of contrastive VL representation learning with multiple captions per image.

6.2.1 Preliminaries

Notation. We closely follow the notation from Chapter 5 and (Bleeker et al., 2023b). See Table 6.A.1 for an overview. Let $\mathcal{D}$ be a dataset of N image-caption tuples: $\mathcal{D} = \left\{ \left(\mathbf{x}_{\mathcal{I}}^i, \{\mathbf{x}_{\mathcal{C}_j}^i\}_{j=1}^k \right) \right\}_{i=1}^N$. Each tuple $i \in N$ contains one image $\mathbf{x}_{\mathcal{I}}^i$ and k captions $\mathbf{x}_{\mathcal{C}_j}^i$, where $1 \leq j \leq k$. All captions in tuple $i \in N$ are considered as matching captions w.r.t. image $\mathbf{x}_{\mathcal{I}}$ in the tuple i. The latent representation of an image-caption pair from a tuple i is denoted as $\mathbf{z}_{\mathcal{I}}^i$ and $\mathbf{z}_{\mathcal{C}_j}^i$ respectively. During training, we sample image-caption pairs from the dataset $\mathcal{D}$ and optimize for the evaluation task T. We include all captions in the dataset once per training epoch, hence, each image is sampled k times.

Given an image $\mathbf{x}_{\mathcal{I}}$, a set of k associated captions $K = \{\mathbf{x}_{\mathcal{C}_j}\}_{j=1}^{k}$, and one caption randomly sampled from the set $\mathbf{x}_{\mathcal{C}} \in K$, we define the following representations:

(i) $\mathbf{z}_{\mathcal{C}\to\mathcal{I}}^{SUF}$ as *sufficient* representation of the caption $\mathbf{x}_{\mathcal{C}}$ that describes the image $\mathbf{x}_{\mathcal{I}}$;

(ii) $\mathbf{z}_{\mathcal{I}\to\mathcal{C}}^{SUF}$ as representation of the image $\mathbf{x}_{\mathcal{I}}$ *sufficient for the caption* $\mathbf{x}_{\mathcal{C}}$;

(iii) $\mathbf{z}_{\mathcal{I}\to\mathcal{C}}^{MIN}$ as representation of the image $\mathbf{x}_{\mathcal{I}}$ that is *minimally sufficient for the caption* $\mathbf{x}_{\mathcal{C}}$; and

(iv) $\mathbf{z}_{\mathcal{I}\to K}^{OPT}$ as representation of the image $\mathbf{x}_{\mathcal{I}}$ that is *optimal for the set of captions* K given the task T.

In addition, we write S_{SynSC} for a synthetic shortcut, S for the original shared information, i.e., information that does not contain synthetic shortcuts, S^+ for the shared information that includes a synthetic shortcut, and R^+ for task-relevant information that contains a synthetic shortcut. In the context of task relevance, we define R and $\neg R$ as task-relevant and task-irrelevant information, respectively, and C as task-relevant information specific for caption $\mathbf{x}_{\mathcal{C}}$.

Setup. We work with a dual-encoder setup, with an image encoder and a caption encoder that do not share parameters. The *image encoder* $f_\theta(\cdot)$ takes an image $\mathbf{x}_{\mathcal{I}}$ as input and returns its latent representation: $\mathbf{z}_{\mathcal{I}} := f_\theta(\mathbf{x}_{\mathcal{I}})$. Similarly, the *caption encoder* $g_\phi(\cdot)$ takes a caption $\mathbf{x}_{\mathcal{C}}$ as input, and encodes the caption into a latent representation: $\mathbf{z}_{\mathcal{C}} := g_\phi(\mathbf{z}_{\mathcal{C}})$. Both $\mathbf{z}_{\mathcal{C}}$ and $\mathbf{z}_{\mathcal{I}}$ are unit vectors projected into d-dimensional multi-modal space: $\mathbf{z}_{\mathcal{C}} \in \mathbb{R}^d$, $\mathbf{z}_{\mathcal{I}} \in \mathbb{R}^d$. For an overview of notation, we refer to Appendix 6.A, Table 6.A.1.

Assumptions. Given an image-caption tuple, we assume that each caption in the tuple is distinct from the other captions in the tuple. We also assume that each caption in the tuple contains two types of task-relevant information:

(i) shared information, i.e., information shared with other captions in the same tuple, and

(ii) caption-specific information, i.e., information that is not shared with the other captions.

For simplicity, we base our subsequent analysis on tuples where one image $\mathbf{x}_{\mathcal{I}}$ is associated with two captions $\mathbf{x}_{\mathcal{C}_A}$ and $\mathbf{x}_{\mathcal{C}_B}$: $\left(\mathbf{x}_{\mathcal{I}}, \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}\right)$. However, the analysis described in this section can be extended to a case with more than two captions. We treat images and captions as views and define $\mathbf{x}_{\mathcal{I}}$, $\mathbf{x}_{\mathcal{C}_A}$, and $\mathbf{x}_{\mathcal{C}_B}$ to be random variables of an image and two matching captions, with the

joint distribution $p(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B})$. For more details on assumptions and problem definition, we refer to Appendix 6.B.

6.2.2 *Analysis of contrastive vision-language representation learning for multiple captions per image*

InfoMax. We start our analysis of contrastive VL representation learning by introducing the InfoMax optimization objective, a typical loss for VL representation learning. The goal of an InfoMax optimization objective, e.g., InfoNCE (van den Oord et al., 2018), is to maximize the mutual information (MI) between the latent representations of two views of the same data (Tschannen et al., 2020). Therefore, the optimization objective is equivalent to: $\max_{f_\theta, g_\phi} I(\mathbf{z}_{\mathcal{I}}; \mathbf{z}_{\mathcal{C}})$ where $\mathbf{z}_{\mathcal{I}} = f_\theta(\mathbf{x}_{\mathcal{I}})$ and $\mathbf{z}_{\mathcal{C}} := g_\phi(\mathbf{x}_{\mathcal{C}})$.

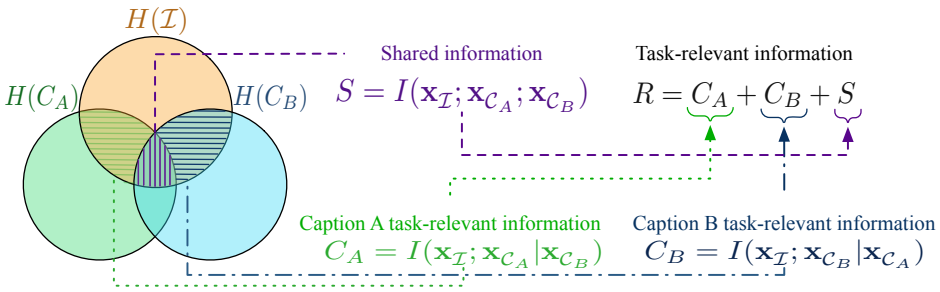


Figure 6.2: We define $H(\mathbf{x}_{\mathcal{I}})$ as image information, $H(\mathbf{x}_{\mathcal{C}_A})$ and $H(\mathbf{x}_{\mathcal{C}_B})$ as caption information; both captions only describe the information depicted in the image and contain shared and caption-specific information. We further define $C_A = I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A} \mid \mathbf{x}_{\mathcal{C}_B})$ and $C_B = I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_B} \mid \mathbf{x}_{\mathcal{C}_A})$ as caption-specific information; $S = I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A}; \mathbf{x}_{\mathcal{C}_B})$ as shared information; $\neg R = H(\mathbf{x}_{\mathcal{I}} \mid \mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B})$ as task-irrelevant information; $R = C_A + C_B + S$ as task-relevant information.

Minimally sufficient image representation. During training, batches of image-caption pairs are sampled. The optimization involves maximizing the MI between the image representation $\mathbf{z}_{\mathcal{I}}$ and the matching caption representation $\mathbf{z}_{\mathcal{C}}$. Wang et al. (2022a) argue that, since all supervision information for one view (i.e., the image) comes from the other view (i.e., the caption), the representations learned contrastively are approximately minimally sufficient. Following (Tian et al., 2020b; Wang et al., 2022a), we extend the definition of sufficient representation to VL context and define sufficient caption representations, sufficient image representations, and minimally sufficient image representation.

Definition 6.2.1 (Sufficient caption representation)**.** *Given an image* $\mathbf{x}_\mathcal{I}$*, and a set of matching captions* $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$*, the representation* $\mathbf{z}^{SUF}_{\mathcal{C}\to\mathcal{I}}$ *of caption* $\mathbf{x}_\mathcal{C} \in \mathcal{C}$ *is sufficient for image* $\mathbf{x}_\mathcal{I}$ *if, and only if,* $I(\mathbf{z}^{SUF}_{\mathcal{C}\to\mathcal{I}}; \mathbf{x}_\mathcal{I}) = I(\mathbf{x}_\mathcal{C}; \mathbf{x}_\mathcal{I})$*.*

The sufficient caption representation $\mathbf{z}^{SUF}_{\mathcal{C}\to\mathcal{I}}$ contains all the information about image $\mathbf{x}_\mathcal{I}$ in caption $\mathbf{x}_\mathcal{C}$.

Definition 6.2.2 (Sufficient image representation)**.** *Given an image* $\mathbf{x}_\mathcal{I}$*, and a set of matching captions* $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$*, the representation* $\mathbf{z}^{SUF}_{\mathcal{I}\to\mathcal{C}}$ *of image* $\mathbf{x}_\mathcal{I}$ *is sufficient for caption* $\mathbf{x}_\mathcal{C} \in \mathcal{C}$ *if, and only if,* $I(\mathbf{z}^{SUF}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_\mathcal{C}) = I(\mathbf{x}_\mathcal{I}; \mathbf{x}_\mathcal{C})$*.*

Similarly, the sufficient image representation $\mathbf{z}^{SUF}_{\mathcal{I}\to\mathcal{C}}$ contains all the shared information between an image $\mathbf{x}_\mathcal{I}$ and a caption $\mathbf{x}_\mathcal{C}$. Note that a sufficient image representation can be sufficient w.r.t. multiple captions.

Definition 6.2.3 (Minimally sufficient image representation)**.** *Given an image* $\mathbf{x}_\mathcal{I}$*, and a set of matching captions* $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$*, the sufficient image representation* $\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}}$ *of image* $\mathbf{x}_\mathcal{I}$ *is minimally sufficient for caption* $\mathbf{x}_\mathcal{C} \in \mathcal{C}$ *if, and only if,* $I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_\mathcal{I}) \leq I(\mathbf{z}^{SUF}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_\mathcal{I})$*, for all* $\mathbf{z}^{SUF}_{\mathcal{I}\to\mathcal{C}}$ *that are sufficient.*

Intuitively, $\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}}$ comprises the smallest amount of information about $\mathbf{x}_\mathcal{I}$ (while still being sufficient) and, therefore, only contains the information that is shared with caption $\mathbf{x}_\mathcal{C}$, i.e., the non-shared information is suppressed.

Task-optimal image representation. The definition of task-optimal image representation is based on the notion of task-relevant information. In the context of VL representation learning with multiple captions per image, we define task-relevant information as all information described by the matching captions. That includes both caption-specific and shared information. Consequently, the task-optimal image representation is image representation that is sufficient w.r.t. all matching captions.

Formally, following assumptions from Appendix 6.B.2, we define the task-relevant information R as all the information described by the matching captions. The task-relevant information can be expressed as follows:

$$\begin{aligned} \underbrace{R}_{\substack{\text{Task-relevant}\\ \text{information}}} &= \underbrace{H(\mathbf{x}_\mathcal{I})}_{\substack{\text{Image}\\ \text{information}}} - \underbrace{H(\mathbf{x}_\mathcal{I} \mid \mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B})}_{\substack{\text{Task-irrelevant}\\ \text{information}}} \\ &= \underbrace{I(\mathbf{x}_\mathcal{I}; \mathbf{x}_{\mathcal{C}_A} \mid \mathbf{x}_{\mathcal{C}_B})}_{\substack{C_A\text{-specific}\\ \text{task-relevant information}}} + \underbrace{I(\mathbf{x}_\mathcal{I}; \mathbf{x}_{\mathcal{C}_B} \mid \mathbf{x}_{\mathcal{C}_A})}_{\substack{C_B\text{-specific}\\ \text{task-relevant information}}} + \underbrace{I(\mathbf{x}_\mathcal{I}; \mathbf{x}_{\mathcal{C}_A}; \mathbf{x}_{\mathcal{C}_B})}_{\substack{\text{Shared}\\ \text{information}}}. \end{aligned} \tag{6.1}$$

Similarly, the task-irrelevant information $\neg R$ is the image information not described by the captions. Figure 6.2 illustrates both definitions.

The multi-view assumption states that the task-relevant information for the downstream tasks comes from the information shared between views (Shwartz-Ziv and LeCun, 2023). However, in the case of VL representation learning with multiple captions per image, the task-relevant information R includes both shared information S, and caption-specific information C_A and C_B (Eq. 6.1).

Definition 6.2.4 (Task-optimal image representation)**.** *Given an image $\mathbf{x}_\mathcal{I}$, and a set of matching captions $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$, the representation $\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}}$ is task-optimal image representation for all matching captions if, and only if, $I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_\mathcal{C}) = I(\mathbf{x}_\mathcal{I}; \mathbf{x}_\mathcal{C})$, for all $\mathbf{x}_\mathcal{C} \in \mathcal{C}$.*

In other words, task-optimal image representations contain all the information that the image shares with the matching captions. Hence, a task-optimal image representation is sufficient w.r.t. all matching captions. The information contained in the task-optimal image representation includes both shared and caption-specific information. Therefore, a task-optimal image representation can never be a minimally sufficient image representation w.r.t. to a specific caption.

Theorem 6.1 (Suboptimality of contrastive learning with multiple captions per image)**.** *Given an image $\mathbf{x}_\mathcal{I}$, a set of matching captions $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$, and a contrastive learning loss function $\mathcal{L}_{InfoNCE}$ that optimizes for task T, image representations learned during contrastive learning will be minimally sufficient and will never be task-optimal image representations.*

The proof is provided in Appendix 6.C. Rephrasing Theorem 6.1, given an image and two captions that form two image-caption pairs, $(\mathbf{x}_\mathcal{I}, \mathbf{x}_{\mathcal{C}_A})$ and $(\mathbf{x}_\mathcal{I}, \mathbf{x}_{\mathcal{C}_B})$, and assuming that contrastive loss optimizes the image encoder to be minimally sufficient w.r.t. to caption $\mathbf{x}_{\mathcal{C}_A}$ during a training step, all task-relevant information C_B specific to caption $\mathbf{x}_{\mathcal{C}_B}$ will be suppressed in $\mathbf{z}_\mathcal{I}$. Hence, the resulting image representation will not be optimal for the task T.

Theorem 6.1 indicates a discrepancy between minimally sufficient representations learned during contrastive training with the InfoNCE loss and the task-optimal image representations in the context of learning VL representations with multiple captions per image. Although the InfoMax loss does not have an explicit constraint to compress information, prior work indicates that feature suppression is happening (Robinson et al., 2021; Shwartz-Ziv and LeCun,

2023). Hence, we question if contrastive loss can be used to learn task-optimal image representations in the context of multiple captions per image.

Furthermore, Theorem 6.1 implies that in the context of contrastive VL representation learning with multiple captions per image, the minimally sufficient representation, which discards non-shared information, is not the same as the task-optimal representation that comprises both caption-specific and shared information. This suggests that the features learned during contrastive learning might be shortcuts, i.e., easy-to-detect discriminatory features that minimize the contrastive optimization objective but are not necessarily sufficient for solving the evaluation task. To examine this problem, we introduce a synthetic shortcuts framework that allows us to investigate the problem of the suboptimality of contrastive learning with multiple captions per image in a controlled way.

6.3 SYNTHETIC SHORTCUTS TO CONTROL SHARED INFORMATION

In Section 6.2 we show the suboptimality of the contrastive InfoNCE loss with multiple captions per image. In the case of real-world VL datasets with multiple captions per image, there are no annotations that indicate the information shared between the image and captions and the information specific to each caption. Hence, we cannot directly measure how much of the shared and unique information is captured by the representations.

Synthetic shortcuts. In this section, we introduce the *synthetic shortcuts for vision-language (SVL)* training and evaluation framework. We denote the *synthetic shortcuts for image-caption data* as S_{SynSC}. The purpose of the framework is to introduce additional and easily identifiable information shared between an image and the matching captions that does not bear any semantic meaning. The shortcuts we use in this chapter are represented as numbers that we add to images and captions. For images, we add the shortcut number by adding MNIST images as an overlay to the original images. For captions, we append the numbers of the shortcut as extra tokens at the end of the caption. Some examples of image-caption pairs with added shortcuts can be seen in Figure 6.D.1.

If contrastive losses learn task-optimal representations, then the presence of synthetic shortcuts should not negatively impact the evaluation performance, since synthetic shortcuts represent additional information and the remaining task-relevant information is intact. By incorporating synthetic shortcuts into the image-caption dataset, the shared information would include the information that was originally shared and the synthetic shortcut: $S^+ = S + S_{SynSC}$. Hence, the task-relevant information would comprise caption-specific information that was originally shared and a synthetic shortcut: $R^+ = C_A + C_B + S + S_{SynSC}$. If injecting a synthetic shortcut influences the performance negatively, we can conclude that by learning to represent a synthetic shortcut the model suppresses other task-relevant information in favor of the shortcut, hence the representation is not task-optimal. The setup is inspired by the "datasets with explicit and controllable competing features," introduced by (Chen et al., 2021a), but we adapt this setup to the VL scenario.

For experiments, we use the Flickr30k and MS-COCO image-caption datasets, that consist of image-caption tuples, each image is associated with five captions. During training, we sample a batch of image-caption pairs $\mathcal{B} = \{(\mathbf{x}^i_{\mathcal{I}}, \mathbf{x}^i_{\mathcal{C}_j}), \dots\}_{i=1}^{|\mathcal{B}|}$, from dataset $\mathcal{D}$, and apply shortcut sampling. We inject the shortcuts in a manner that preserves the original information of the images and captions. Furthermore, we append the shortcut after applying data augmentations to ensure that the shortcut is present in both the images and captions (i.e., the shortcut is not augmented away). We refer to Figure 6.D.1 for some examples. The training, evaluation, and implementation details of the shortcut sampling are provided in Appendix 6.D.4.

We define the following experimental setups:

I *No shortcuts*: As a baseline, we fine-tune a pre-trained CLIP (Radford et al., 2021) and train VSE++ (Faghri et al., 2018) from scratch on Flickr30k and MS-COCO, without using any shortcuts. The experimental setup for training both models is provided in Appendix 6.D.2 and 6.D.3. The goal of this setup is to show the retrieval evaluation performance without adding any shortcuts for both a large-scale pre-trained foundation model and a small-scale model trained from scratch.

II *Unique shortcuts*: We add a unique shortcut to each image-caption tuple $i \in \mathcal{D}$ in the dataset. In this setup, each image caption pair can be uniquely matched during training by only detecting the shortcut. For each tuple $i \in \mathcal{D}$, we use the number i as the number of the shortcut

we inject to the image and captions in the tuple. If the contrastive loss learns task-optimal representations, the downstream evaluation performance should not decrease when training with unique shortcuts.

III *Unique shortcuts on only one modality*: To show that the shortcuts do not interfere with the original task-relevant information (S, C_A, and C_B) of the images and captions, we create a dataset with only shortcuts on either the image or caption modality. Therefore, the shortcut cannot be used by the encoders to match an image-caption pair. Hence, we expect the encoders to ignore the shortcuts and extract the features from the original data similar to the features learned by the baseline models in experimental setup I.

IV *N bits of shortcuts*: In this setup, for each image-caption pair in the training batch $\mathcal{B}$, we randomly sample a shortcut number from the range $[0, 2^n]$, where n is the number of bits. The higher the value of n, the more image-caption pairs in the training batch will have by expectation a unique shortcut, and, the less the model has to rely on S and the remaining task-relevant information to solve the contrastive objective. The goal of this setup is to show that, the more unique (shortcut) information is present per sample in the batch, the less contrastive models rely on the remaining task-relevant information.

It should be noted that the shortcuts we add are independent of the image-caption pairs. However, the goal of the SVL framework is to measure the effect of the presence of additional easy-to-detect shared information on the learned representations.

Evaluation method. To show the effect of the injected shortcuts on retrieval evaluation performance, we evaluate both with and without adding the shortcuts during evaluation. When training with unique shortcuts, we add a unique shortcut to each tuple in the test set as well. When training with shortcuts on either one of the two modalities, we only evaluate without shortcuts to show that training with shortcuts on one modality does not influence performance. When training with n bits of shortcuts, we add the shortcut $\mod(i, n)$ (modulo) to each tuple i in the evaluation set, to make sure we use the same number of shortcuts during evaluation as during training. To facilitate the reproducibility of our work and support further research, we provide the code with this chapter.[2]

2 https://github.com/MauritsBleeker/svl-framework

6.4 SYNTHETIC SHORTCUTS AND THEIR IMPACT ON THE LEARNED REPRESENTATIONS AND EVALUATION PERFORMANCE

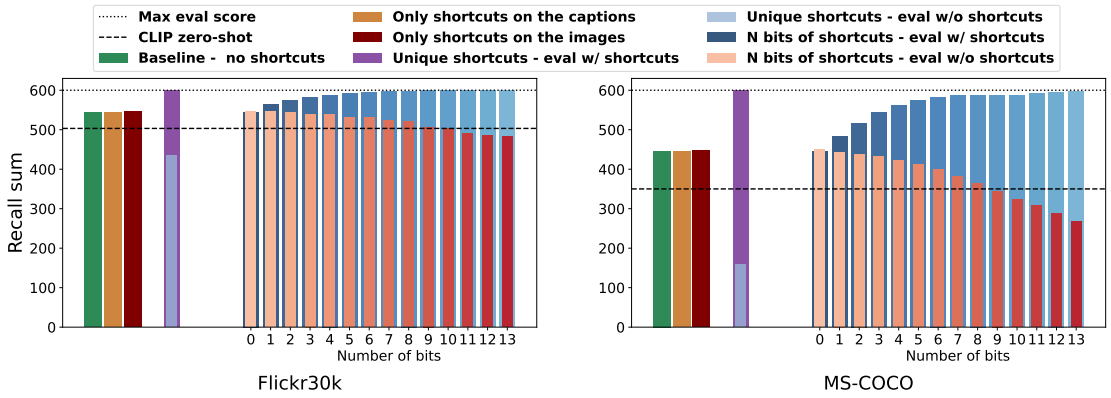
6.4.1 Findings

First, we train and evaluate both a CLIP and VSE++ without shortcuts on the Flickr30k and MS-COCO dataset for the image-caption retrieval task as a baseline. We use the recall sum (i.e., the sum of $R@1$, $R@5$, and $R@10$ for both i2t and t2i retrieval) as evaluation metric (see Appendix 6.B.1 for the evaluation task description). We visualize the results in Figure 6.3. The dotted line (in Figure 6.3a and 6.3b) indicates the maximum evaluation score (i.e., 600). For CLIP, we also provide the zero-shot performance of the model, indicated by the dashed line in Figure 6.3a. When referring to specific results in Figure 6.3, we use the color of the corresponding bar and legend key in brackets in the text.

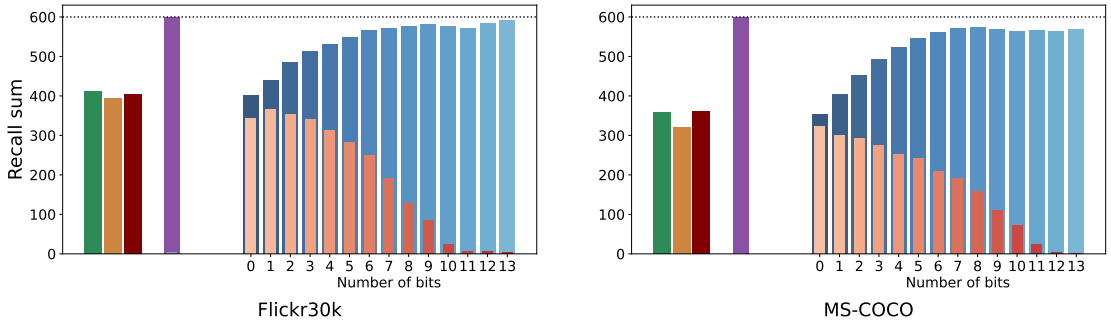
Based on Figure 6.3, we draw the following conclusions:

I When training CLIP and VSE++ with only shortcuts on either the caption modality (in Figure 6.3, the corresponding bar/legend box is colored) or on the image modality (, in Figure 6.3), we do not observe a drop in evaluation scores for CLIP compared to the baseline model (, in Figure 6.3a). For VSE++ we only observe a slight drop in evaluation score when training with shortcuts on the caption modality (again , mainly for MS-COCO, in Figure 6.3b). Therefore, we conclude that the synthetic shortcuts do not interfere with the original shared information S or other task-relevant information.

II When training the models with *unique shortcuts*, we observe for both CLIP and VSE++ that when evaluating with shortcuts (, in Figure 6.3), the models obtain a perfect evaluation score. When evaluating without shortcuts (, in Figure 6.3) the evaluation score for VSE++ drops to zero and for CLIP below the zero-shot performance. We conclude that with unique shortcuts: (i) both CLIP and VSE++ fully rely on the shortcuts to solve the evaluation task, (ii) VSE++ has not learned any other shared or task-relevant information other than the shortcuts (since it is trained from scratch, only detecting the shortcuts is sufficient to minimize the



(a) Evaluation scores for CLIP, applying different setups of shortcut sampling.



(b) Evaluation scores for VSE++, appling different setups of shortcut sampling.

Figure 6.3: The effect of synthetic shortcuts on the performance of CLIP and VSE++, when evaluated on the ICR task. The dotted line indicates the maximum evaluation score for the recall sum. For CLIP we indicate with the dashed line the zero-shot evaluation performances. (Best viewed in color.)

contrastive loss), and (iii) fine-tuned CLIP has suppressed original features from the zero-shot model in favor of the shortcuts.

III When training the models with *N bits of shortcuts*, we observe for both CLIP and VSE++ that the larger the number of bits we use during training and when evaluating without shortcuts (, in Figure 6.3), the bigger the drop in evaluation performance. When we evaluate with shortcuts (, in Figure 6.3), the evaluation performance improves as we use more bits compared to the baseline without shortcuts , in Figure 6.3). For VSE++, evaluating without shortcuts (, in Figure 6.3b) results in a drop to zero when having a large number of bits. For CLIP, the evaluation performance drops below the zero-shot performance. If we train with 0 bits of shortcuts (i.e., the shortcut is a constant) we do not observe any drop or increase in evaluation scores for CLIP.

6.4.2 Upshot

Given the findings based on Figure 6.3 we conclude that a contrastive loss (i.e., InfoNCE) mainly learns the easy-to-detect minimal shared features among image-caption pairs that are sufficient to minimize the contrastive objective while suppressing the remaining shared and/or task-relevant information. If contrastive losses are sufficient to learn task-optimal representations for image-caption matching, these shortcuts should not adversely impact the evaluation performance. Moreover, if the contrastive loss would only learn features that are shared among the image and all captions (i.e, S), we should not observe a drop in performance to 0 for the VSE++ model when training with unique shortcuts, since there is still a lot of task-relevant information present in S. Especially in a training setup where a model is trained from scratch or fine-tuned on small datasets, the easy-to-detect features are likely not equivalent to all task-relevant information in the images and captions. Hence, we conclude that the contrastive loss itself is not sufficient to learn task-optimal representations of the images (and sufficient representations of captions) and that it only learns the minimal easy-to-detect features that are needed to minimize the contrastive objective.

6.5 REDUCING SHORTCUT LEARNING

In Section 6.4 we have shown that the contrastive loss mainly relies on the minimal, easy-to-detect features shared among image-caption pairs while suppressing remaining task-relevant information. In this section, we describe two methods that help to reduce shortcut learning for contrastive learning on our SVL framework: latent target decoding (Bleeker et al., 2023b) and implicit feature modification (Robinson et al., 2021). In Section 6.6, we present the evaluation results.

6.5.1 Latent target decoding

Latent target decoding (LTD) (Bleeker et al., 2023b) is a method to reduce predictive feature suppression (i.e., shortcut learning) for resource-constrained contrastive image-caption matching. The contrastive objective (i..e, InfoNCE) is

combined with an additional reconstruction loss, which reconstructs the input caption from the latent representation of the caption $\mathbf{z}^i_{\mathcal{C}_j}$. Instead of reconstructing the tokens of the input caption in an auto-regressive manner (i.e., auto-encoding), the caption is reconstructed non-auto-regressively, by mapping the caption representation into the latent space of a Sentence-BERT (Reimers and Gurevych, 2019; Song et al., 2020) and minimizing the distance (i.e., reconstructing) between the reconstruction and the Sentence-BERT representation of the caption $\mathbf{x}^i_{\mathcal{C}_j}$. The assumption is that the *target* generated by the Sentence-BERT model contains all task-relevant information in the caption. Hence, by correctly mapping the latent caption representation $\mathbf{z}^i_{\mathcal{C}_j}$ into the latent space of Sentence-BERT, the caption encoder cannot suppress any task-relevant information or rely on shortcut solutions. LTD is implemented both as a dual-loss objective (i.e., the contrastive loss and LTD are added up) and as an optimization constraint while minimizing the InfoNCE loss, by implementing the loss as a Lagrange multiplier.

Experimental setup. We use the LTD implementation and setup similar to Bleeker et al. (2023b). We train both CLIP and VSE++ with LTD, implemented as either dual loss or an optimization constraint. When implementing LTD as a constraint, we try $\eta \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$ as bound values. Similar to (Bleeker et al., 2023b), when implementing LTD as a dual loss, we use $\beta = 1$ as balancing parameters. We train both with and without unique shortcuts. We do this to show (i) what the performance improvement is compared to using only InfoNCE, and (ii) to what degree LTD prevents full collapse to shortcut features. For each model and dataset, we take the training setup that results in the highest performance on the validation set.

6.5.2 Implicit feature modification

Implicit feature modification (IFM) (Robinson et al., 2021) is a method, originally introduced for contrastive representation learning for images, that applies perturbations to features learned by a contrastive loss (InfoNCE). IFM perpetuates features that the encoders use during a training step to discriminate between positive and negative samples. By doing so, it removes some of the features that are currently used to solve the discrimination task, to avoid the InfoNCE loss to learn shortcut solutions. How much of the features are

removed, is defined by a perturbation budget ϵ. IFM is implemented as a dual loss in combination with the InfoNCE loss.

Experimental setup. We apply a similar experimental setup for IFM as for LTD. We apply IFM both to CLIP and to VSE++, both with and without unique shortcuts. Similar to (Robinson et al., 2021), we try different permutation budgets ϵ, we try $\epsilon \in \{0.05, 0.1, 0.2, 0.5, 1\}$. In line with the LTD setup, we take the training setup that results in the highest performance on the validation set.

6.6 EXPERIMENTAL RESULTS

6.6.1 Does latent target decoding reduce shortcut learning?

In Table 6.1 we summarize the effect of LTD on reducing shortcut learning.

For CLIP, for both the Flickr30k and MS-COCO dataset, we do not observe an increase in recall scores when fine-tuning with $\mathcal{L}_{\text{InfoNCE+LTD}}$ compared to models that are only fine-tuned with $\mathcal{L}_{\text{InfoNCE}}$. LTD has originally been proposed for resource-constrained VL models. We argue that the additional features that LTD can extract are either already present in the pre-trained CLIP model, or not relevant for the evaluation task. However, when fine-tuning with $\mathcal{L}_{\text{InfoNCE+LTD}}$ and in the presence of shortcuts in the training data, degradation in recall scores is significantly lower than when fine-tuned only with the $\mathcal{L}_{\text{InfoNCE}}$. This shows that LTD can reduce the suppression of features in favor of the shortcut features when fine-tuning large-scale VL models.

Across the board, VSE++ models trained with the $\mathcal{L}_{\text{InfoNCE+LTD}}$ loss consistently outperform the $\mathcal{L}_{\text{InfoNCE}}$ loss, both for i2t and t2i retrieval and both when trained either with or without shortcuts, as indicated by higher recall@k scores; this is consistent with the findings presented in (Bleeker et al., 2023b). For both the Flickr30k and MS-COCO dataset, when trained with the $\mathcal{L}_{\text{InfoNCE}}$ and with shortcuts present in the training data, the model performance collapses to around 0 in the absence of shortcuts (as we have seen in Section 6.4). However, when we train with shortcuts in the training data and with $\mathcal{L}_{\text{InfoNCE+LTD}}$, we observe, for both Flickr30k and MS-COCO, a significant gain in performance. The performance improvement is bigger for Flickr30k than for MS-COCO. In general, the recall scores are still significantly lower than training without shortcuts, however, the models do not solely rely on the shortcuts anymore to min-

Table 6.1: Mean and variance (over three training runs) recall@k evaluation scores for the Flickr30k and MS-COCO datasets for image-to-text and text-to-image retrieval. We train with two loss functions: $\mathcal{L}_{\text{InfoNCE}}$ and $\mathcal{L}_{\text{InfoNCE+LTD}}$. We train either with (✓) or without (✗) shortcuts. For the model trained with $\mathcal{L}_{\text{InfoNCE+LTD}}$, we provide the hyper-parameters of the best-performing model. η indicates that the best-performing model uses LTD implemented as an optimization constraint with bound η. β indicates that the best-performing model uses LTD implemented as a dual-loss with $\beta = 1$.

		i2t			*t2i*			
Loss	S_{SynSC}	R@1	R@5	R@10	R@1	R@5	R@10	rsum
Flickr30k								
		CLIP						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$86.9_{\pm 0.1}$	$\mathbf{97.4}_{\pm 0.1}$	$\mathbf{99.0}_{\pm 0.0}$	$72.4_{\pm 0.1}$	$\mathbf{92.1}_{\pm 0.0}$	$\mathbf{95.8}_{\pm 0.0}$	$543.5_{\pm 1.1}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\beta = 1$	✗	$86.5_{\pm 0.6}$–	$97.1_{\pm 0.0}\downarrow$	$98.5_{\pm 0.0}\downarrow$	$72.4_{\pm 0.0}$–	$92.3_{\pm 0.0}\downarrow$	$95.9_{\pm 0.0}\downarrow$	$542.8_{\pm 0.8}$–
$\mathcal{L}_{\text{InfoNCE}}$	✓	$57.2_{\pm 8.3}$	$84.0_{\pm 4.8}$	$91.0_{\pm 1.9}$	$44.9_{\pm 4.5}$	$74.9_{\pm 6.0}$	$84.2_{\pm 2.5}$	$436.2_{\pm 145.0}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\beta = 1$	✓	$\mathbf{64.0}_{\pm 1.3}\uparrow$	$\mathbf{87.8}_{\pm 0.9}\uparrow$	$\mathbf{93.2}_{\pm 0.8}\uparrow$	$\mathbf{50.7}_{\pm 0.6}\uparrow$	$\mathbf{79.8}_{\pm 0.7}\uparrow$	$\mathbf{88.1}_{\pm 0.5}\uparrow$	$\mathbf{463.6}_{\pm 17.3}\uparrow$
		VSE++						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$52.6_{\pm 1.1}$	$79.8_{\pm 0.1}$	$87.8_{\pm 0.1}$	$39.5_{\pm 0.3}$	$69.8_{\pm 0.0}$	$79.4_{\pm 0.1}$	$409.0_{\pm 4.0}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\eta = 0.2$	✗	$\mathbf{54.1}_{\pm 0.1}\uparrow$	$\mathbf{81.1}_{\pm 0.8}\uparrow$	$\mathbf{88.6}_{\pm 0.1}\uparrow$	$\mathbf{42.5}_{\pm 0.0}\uparrow$	$\mathbf{71.9}_{\pm 0.1}\uparrow$	$\mathbf{81.3}_{\pm 0.0}\uparrow$	$\mathbf{419.6}_{\pm 0.1}\uparrow$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$0.1_{\pm 0.0}$	$0.6_{\pm 0.1}$	$1.1_{\pm 0.1}$	$0.1_{\pm 0.0}$	$0.5_{\pm 0.0}$	$1.0_{\pm 0.0}$	$3.4_{\pm 0.6}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\eta = 0.05$	✓	$\mathbf{24.7}_{\pm 0.5}\uparrow$	$\mathbf{51.8}_{\pm 0.7}\uparrow$	$\mathbf{65.6}_{\pm 1.4}\uparrow$	$\mathbf{20.7}_{\pm 1.0}\uparrow$	$\mathbf{49.2}_{\pm 0.6}\uparrow$	$\mathbf{62.6}_{\pm 1.2}\uparrow$	$\mathbf{274.6}_{\pm 4.6}\uparrow$
MS-COCO								
		CLIP						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$63.8_{\pm 0.3}$	$86.1_{\pm 0.2}$	$92.3_{\pm 0.0}$	$46.3_{\pm 0.3}$	$74.8_{\pm 0.1}$	$84.1_{\pm 0.2}$	$447.5_{\pm 0.5}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\beta = 1$	✗	$63.8_{\pm 0.0}$–	$86.1_{\pm 0.0}$–	$92.3_{\pm 0.0}$–	$46.3_{\pm 0.0}$–	$74.7_{\pm 0.0}$–	$84.1_{\pm 0.0}$–	$447.4_{\pm 0.0}$–
$\mathcal{L}_{\text{InfoNCE}}$	✓	$13.6_{\pm 0.9}$	$31.5_{\pm 2.4}$	$42.2_{\pm 3.7}$	$7.3_{\pm 0.6}$	$22.1_{\pm 1.0}$	$32.7_{\pm 1.7}$	$149.4_{\pm 32.7}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\beta = 1$	✓	$\mathbf{18.9}_{\pm 0.1}\uparrow$	$\mathbf{41.8}_{\pm 0.1}\uparrow$	$\mathbf{54.1}_{\pm 0.1}\uparrow$	$\mathbf{16.5}_{\pm 0.0}\uparrow$	$\mathbf{39.4}_{\pm 0.0}\uparrow$	$\mathbf{52.6}_{\pm 0.1}\uparrow$	$\mathbf{223.4}_{\pm 0.2}\uparrow$
		VSE++						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$42.2_{\pm 0.1}$	$72.7_{\pm 0.1}$	$83.2_{\pm 0.1}$	$30.9_{\pm 0.0}$	$61.2_{\pm 0.1}$	$73.5_{\pm 0.1}$	$363.8_{\pm 2.3}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\eta = 0.1$	✗	$\mathbf{43.6}_{\pm 0.1}\uparrow$	$\mathbf{73.5}_{\pm 0.0}\uparrow$	$\mathbf{83.7}_{\pm 0.0}\uparrow$	$\mathbf{32.4}_{\pm 0.1}\uparrow$	$\mathbf{62.5}_{\pm 0.0}\uparrow$	$\mathbf{74.7}_{\pm 0.0}$	$\mathbf{370.5}_{\pm 0.1}\uparrow$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$0.0_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.2_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.2_{\pm 0.0}$	$0.7_{\pm 0.0}$
$\mathcal{L}_{\text{InfoNCE+LTD}}$, $\eta = 0.01$	✓	$\mathbf{3.9}_{\pm 0.0}\uparrow$	$\mathbf{13.7}_{\pm 0.6}\uparrow$	$\mathbf{21.6}_{\pm 0.9}\uparrow$	$\mathbf{3.1}_{\pm 0.2}\uparrow$	$\mathbf{11.0}_{\pm 1.6}\uparrow$	$\mathbf{18.1}_{\pm 3.0}\uparrow$	$\mathbf{71.3}_{\pm 3.6}\uparrow$

imize the contrastive loss and are able during evaluation (in the absence of shortcuts) to still correctly match image-caption pairs with each other. The results in Table 6.1 show that LTD is able, in the presence of shortcuts in the training data, to guide (small-scale) VL models that are trained from scratch to not only learn the shortcut features that minimize the contrastive training

Table 6.2: Mean and variance (over three training runs) recall@k evaluation scores for the Flickr30k and MS-COCO datasets for image-to-text and text-to-image retrieval. We train with two loss functions: $\mathcal{L}_{\text{InfoNCE}}$ and $\mathcal{L}_{\text{InfoNCE+IFM}}$. We train either with (✓) or without (✗) shortcuts. For the model trained with $\mathcal{L}_{\text{InfoNCE+IFM}}$, we provide the hyperparameters of the best-performing model.

		i2t			*t2i*			
Loss	S_{SynSC}	R@1	R@5	R@10	R@1	R@5	R@10	rsum
Flickr30k								
		CLIP						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$86.9_{\pm 0.1}$	$97.4_{\pm 0.0}$	$98.8_{\pm 0.0}$	$72.8_{\pm 0.2}$	$92.1_{\pm 0.0}$	$95.6_{\pm 0.0}$	$543.5_{\pm 1.3}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✗	$\mathbf{87.4}_{\pm 0.1}\uparrow$	$97.4_{\pm 0.2}-$	$99.1_{\pm 0.0}-$	$73.2_{\pm 0.0}-$	$92.2_{\pm 0.0}-$	$95.6_{\pm 0.0}-$	$544.9_{\pm 0.2}-$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$57.9_{\pm 0.3}$	$84.6_{\pm 0.8}$	$91.3_{\pm 0.0}$	$43.9_{\pm 2.2}$	$74.6_{\pm 0.8}$	$84.4_{\pm 0.4}$	$436.7_{\pm 18.8}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.1$	✓	$\mathbf{73.8}_{\pm 0.8}\uparrow$	$\mathbf{91.5}_{\pm 0.5}\uparrow$	$\mathbf{95.6}_{\pm 0.0}\uparrow$	$\mathbf{58.9}_{\pm 0.1}\uparrow$	$\mathbf{84.4}_{\pm 0.1}\uparrow$	$\mathbf{91.1}_{\pm 0.2}\uparrow$	$\mathbf{495.2}_{\pm 5.7}\uparrow$
		VSE++						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$\mathbf{52.9}_{\pm 0.2}$	$\mathbf{80.5}_{\pm 0.1}$	$\mathbf{87.6}_{\pm 0.4}$	$\mathbf{40.5}_{\pm 0.1}$	$68.8_{\pm 0.4}$	$\mathbf{78.9}_{\pm 0.3}$	$\mathbf{409.3}_{\pm 2.6}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✗	$52.4_{\pm 0.2}\downarrow$	$76.9_{\pm 0.1}\downarrow$	$85.3_{\pm 0.0}\downarrow$	$39.1_{\pm 0.0}\downarrow$	$68.8-_{\pm 0.1}$	$78.2_{\pm 0.1}\downarrow$	$400.7_{\pm 0.0}\downarrow$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$0.1_{\pm 0.0}$	$0.4_{\pm 0.0}$	$0.8_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.4_{\pm 0.0}$	$1.0_{\pm 0.0}$	$2.9_{\pm 0.0}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✓	$0.0_{\pm 0.0}-$	$0.6_{\pm 0.1}-$	$0.9_{\pm 0.2}-$	$0.1_{\pm 0.0}-$	$0.5_{\pm 0.0}-$	$1.0_{\pm 0.0}-$	$3.2_{\pm 0.8}-$
MS-COCO								
		CLIP						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$\mathbf{63.5}_{\pm 0.1}$	$\mathbf{86.0}_{\pm 0.3}$	$\mathbf{92.2}_{\pm 0.0}$	$46.3_{\pm 0.0}$	$74.7_{\pm 0.0}$	$84.2_{\pm 0.0}$	$446.9_{\pm 0.9}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✗	$63.0_{\pm 0.1}\downarrow$	$86.6_{\pm 0.1}\downarrow$	$92.6_{\pm 0.2}\downarrow$	$\mathbf{47.2}_{\pm 0.0}\uparrow$	$\mathbf{75.6}_{\pm 0.0}\uparrow$	$\mathbf{84.5}_{\pm 0.0}\uparrow$	$\mathbf{449.5}_{\pm 1.7}\uparrow$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$13.9_{\pm 0.0}$	$32.7_{\pm 0.1}$	$43.8_{\pm 0.0}$	$8.8_{\pm 0.0}$	$24.7_{\pm 0.2}$	$35.5_{\pm 0.5}$	$159.4_{\pm 3.4}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✓	$\mathbf{23.4}_{\pm 1.5}\uparrow$	$\mathbf{46.5}_{\pm 2.7}\uparrow$	$\mathbf{58.2}_{\pm 2.5}\uparrow$	$\mathbf{17.1}_{\pm 0.3}\uparrow$	$\mathbf{38.9}_{\pm 0.9}\uparrow$	$\mathbf{51.3}_{\pm 1.0}\uparrow$	$\mathbf{235.5}_{\pm 43.8}\uparrow$
		VSE++						
$\mathcal{L}_{\text{InfoNCE}}$	✗	$\mathbf{41.7}_{\pm 0.3}$	$\mathbf{72.5}_{\pm 0.1}$	$\mathbf{83.1}_{\pm 0.1}$	$\mathbf{31.3}_{\pm 0.0}$	$61.1_{\pm 0.0}$	$73.6_{\pm 0.0}$	$\mathbf{363.4}_{\pm 0.4}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✗	$40.2_{\pm 0.0}\downarrow$	$70.8_{\pm 0.1}\downarrow$	$81.6_{\pm 0.1}\downarrow$	$30.8_{\pm 0.0}\downarrow$	$\mathbf{61.5}_{\pm 0.0}\uparrow$	$\mathbf{74.3}_{\pm 0.0}\uparrow$	$359.3_{\pm 1.1}\downarrow$
$\mathcal{L}_{\text{InfoNCE}}$	✓	$0.0_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.2_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.1_{\pm 0.0}$	$0.2_{\pm 0.0}$	$0.6_{\pm 0.0}$
$\mathcal{L}_{\text{InfoNCE+IFM}}$, $\epsilon = 0.05$	✓	$0.0_{\pm 0.0}-$	$0.1_{\pm 0.0}-$	$0.2_{\pm 0.0}-$	$0.0_{\pm 0.0}-$	$0.1_{\pm 0.0}-$	$0.2_{\pm 0.0}-$	$0.7_{\pm 0.0}-$

objective but also represent other remaining task-relevant features in the data that are not extracted by $\mathcal{L}_{\text{InfoNCE}}$.

6.6.2 *Does implicit feature modification reduce shortcut learning?*

In Table 6.2 we summarize the effect of IFM on reducing shortcut solutions.

For CLIP, we observe that $\mathcal{L}_{\text{InfoNCE+IFM}}$, when training without shortcuts in the training data, only improves performance for the MS-COCO dataset for the t2i task. However, for both Flickr30k and MS-COCO we observe that, when training with unique shortcuts in the training data, fine-tuning with $\mathcal{L}_{\text{InfoNCE+IFM}}$ results in a significantly lower performance drop in recall score than when fine-tuning with the $\mathcal{L}_{\text{InfoNCE}}$. Similar to LTD, the recall@k scores are still lower than when trained without shortcuts in the training data. We conclude that IFM is sufficient to reduce the suppression of features in favor of the shortcut features when fine-tuning a large-scale VL model, as indicated by higher recall@k scores when evaluating without shortcuts.

For VSE++, both for the Flickr30k and MS-COCO dataset, we do not observe that $\mathcal{L}_{\text{InfoNCE+IFM}}$ outperforms the $\mathcal{L}_{\text{InfoNCE}}$, both with and without shortcuts present in the training data. We even observe that $\mathcal{L}_{\text{InfoNCE+IFM}}$, when training without shortcuts, results in a decrease in performance across all recall@k metrics. When training with $\mathcal{L}_{\text{InfoNCE+IFM}}$ and with unique shortcuts in the training data, the evaluation performance still collapses to around 0. The results in Table 6.2 show that IFM is not sufficient to prevent models trained from scratch from fully collapsing to the artificial shortcut solutions we introduce in this work (as opposed to LTD).

6.6.3 Upshot

In this section, we have evaluated two methods for reducing shortcut learning on our SVL framework: LTD and IFM. LTD proves effective in reducing shortcut learning for both CLIP and VSE++. IFM demonstrates its efficacy solely during the fine-tuning of CLIP. These findings indicate that our SVL framework is a challenging and interesting framework to study and evaluate shortcut learning for contrastive VL models. Moreover, our results show that shortcut learning is only partially addressed by the evaluated methods since the evaluation results are not on par with the results on data lacking synthetic shortcuts.

6.7 RELATED WORK

We discuss related work on multi-view representation learning, vision-language learning, and shortcut learning.

6.7.1 Multi-view representation learning

To learn the underlying semantics of the training data, a subgroup of representation learning methods involves training neural encoders that maximize the agreement between representations of similar *views* (van den Oord et al., 2018; Hjelm et al., 2019; Chen et al., 2020c; Radford et al., 2021; Bardes et al., 2022). In general, for uni-modal representation learning, data augmentations are used to generate different views of the same data point. One of the core assumptions in multi-view representation learning is that each view shares the same *task-relevant information* (Sridharan and Kakade, 2008; Zhao et al., 2017; Federici et al., 2020; Tian et al., 2020a; Shwartz-Ziv and LeCun, 2023). However, the optimal view for contrastive self-supervised learning (SSL) (i.e., which information is shared among views/which data augmentation is used) is task-dependent (Tian et al., 2020b; Xiao et al., 2021). Therefore, maximizing the mutual information (MI) between representations of views (i.e., shared information) does not necessarily result in representations that generalize better to down-stream evaluation tasks, since the representations may contain too much additional noise that is irrelevant for the downstream task (Tian et al., 2020b; Tschannen et al., 2020). An open problem in multi-view SSL is to learn representations that contain all task-relevant information from views where each view contains distinct, task-relevant information (Shwartz-Ziv and LeCun, 2023), this is especially a problem in the multi-modal learning domain (Zong et al., 2023).

Chen et al. (2021a) investigate multi-view representation learning for images using contrastive losses. They demonstrate that when multiple competing features exist that redundantly predict the match between two views, contrastive models tend to focus on learning the easy-to-represent features while suppressing other task-relevant information. This results in contrastive losses mainly capturing the easy features, even if all task-relevant information is shared between the two views, suppressing the remaining relevant information.

Several optimization objectives have been introduced to either maximize the lower bound on the MI between views and their latent representations (van den Oord et al., 2018; Bachman et al., 2019; Hjelm et al., 2019; Tian et al., 2020a) or minimize the MI between representations of views while keeping the task-relevant information (Federici et al., 2020; Lee et al., 2021). To learn more task-relevant information that either might not be shared between views or that is compressed by a contrastive loss, several works proposed additional reconstruction objectives to maximize the MI between the latent representation and input data (Tsai et al., 2021; Wang et al., 2022a; Bleeker et al., 2023b; Li et al., 2023b). Liang et al. (2023) introduce a multimodal contrastive objective that factorizes the representations into shared and unique information, while also removing task-irrelevant information by minimizing the upper bound on MI between similar views.

6.7.2 Vision-language representation learning

The goal of VL representation learning is to combine information from the visual and textual modalities into a joint representation or learn coordinated representations (Baltrusaitis et al., 2019; Guo et al., 2019b). The representation learning approaches can be separated into several groups.

Contrastive methods represent one prominent category of VL representation methods. The approaches in this group are typically dual encoders. Early methods in this category are trained from scratch; for instance, (Frome et al., 2013) proposed a VL representation learning model that features a skip-gram language model and a visual object categorization component trained with the hinge rank loss. Another subgroup of methods uses a *dual encoder* with a hinge-based triplet loss (Kiros et al., 2014; Lee et al., 2018; Li et al., 2019a). (Kiros et al., 2014) use the loss for training a CNN-RNN dual encoder. Li et al. (2019a) leverage bottom-up attention and graph convolutional networks (Kipf and Welling, 2017) to learn the relationship between image regions. (Lee et al., 2018) add stacked cross-attention to use both image regions and words as context.

More recently, contrastive approaches involve transformer-based dual encoders trained with more data than the training data from the evaluation set(s). ALBEF (Li et al., 2021) propose to contrastively align unimodal representations before fusion, while X-VLM (Zeng et al., 2022) employs an additional cross-

modal encoder to learn fine-grained VL representations. Florence (Yuan et al., 2021) leverages various adaptation models for learning fine-grained object-level representations. CLIP (Radford et al., 2021), a scaled-up dual encoder, is pre-trained on the task of predicting which caption goes with which image. ALIGN (Jia et al., 2021) uses a simple dual encoder trained on over a billion image alt-text pairs. FILIP (Yao et al., 2022) is a transformer-based bi-encoder that features late multimodal interaction meant to capture fine-grained representations. SLIP (Mu et al., 2022) combines language supervision and image self-supervision to learn visual representations without labels. DeCLIP (Li et al., 2022b) proposes to improve the efficiency of CLIP pretraining using intra-modality self-supervision, cross-modal multi-view supervision, and nearest neighbor supervision.

Another line of work includes learning VL representations using models that are inspired by BERT (Devlin et al., 2019). ViLBERT (Lu et al., 2019) and LXMERT (Tan and Bansal, 2019) expand upon BERT by introducing a two-stream architecture, where two transformers are applied to images and text independently, which is fused by a third transformer in a later stage. B2T2 (Alberti et al., 2019), VisualBERT (Li et al., 2019b), Unicoder-VL (Li et al., 2020a), VL-BERT (Su et al., 2020), and UNITER (Chen et al., 2020d) propose a single-stream architecture, where a single transformer is applied to both images and text. Oscar (Li et al., 2020c) uses caption object tags as anchor points that are fed to the transformer alongside region features. BEIT-3 (Wang et al., 2023) adapt multiway transformers trained using cross-entropy loss (Bao et al., 2022).

Another category of methods for learning VL representations are generative methods, that imply learning VL representation by generating new instances of one modality conditioned on the other modality. For instance, BLIP (Li et al., 2022a) bootstraps captions by generating synthetic captions and filtering out the noisy ones; BLIP-2 (Li et al., 2023a) bootstraps VL representation learning and, subsequently, vision-to-language generative learning. On the other hand, Tschannen et al. (2023) propose to pretrain a encoder-decoder architecture via the image captioning task.

6.7.3 Shortcut learning

Geirhos et al. (2020) define shortcuts in deep neural networks as "decision rules that perform well on standard benchmarks but fail to transfer to more

challenging testing conditions, such as real-world scenarios." In the context of deep learning, a shortcut solution can also be seen as a discrepancy between the features that a model has learned during training and the intended features that a model should learn to perform well during evaluation. For example, shortcuts might be features that minimize the training objective but are much easier to detect than the intended features that are relevant to the evaluation task. Shortcut learning can be caused by biases in the dataset or inductive biases in either the network architecture or training objective.

Hermann and Lampinen (2020) design a dataset with multiple predictive features, where each feature can be used as a label for an image classification task. The authors show that in the presence of multiple features that each redundantly predict the target label, the deep neural model chooses to represent only one of the predictive features that are the easiest to detect, i.e., the model favors features that are easy to detect over features that are harder to discriminate. Next to that, they show that features that are not needed for a classification task, are in general suppressed by the model instead of captured in the learned latent representations.

Robinson et al. (2021) show that contrastive losses can have multiple local minima, where different local minima can be achieved by suppressing features from the input data (i.e., the model learns a shortcut by not learning all task-relevant features). To mitigate the shortcut learning problem, Robinson et al. (2021) propose implicit feature modification, a method that perpetuates the features of positive and negative samples during training to encourage the model to capture different features than the model currently relies on.

Scimeca et al. (2022) design an experimental set-up with multiple shortcut cues in the training data, where each shortcut is equally valid w.r.t. predicting the correct target label. The goal of the experimental setup is to investigate which cues are preferred to others when learning a classification task.

Latent target decoding (LTD) is a method to reduce predictive feature suppression (i.e., shortcuts) for resource-constrained contrastive ICR by reconstructing the input caption in a non-auto-regressive manner. Bleeker et al. (2023b) argue that most of the task-relevant information for the ICR task is captured by the text modality. Hence, the focus is on the reconstruction of the text modality instead of the image modality. Bleeker et al. (2023b) add a decoder to the learning algorithm, to reconstruct the input caption. Instead of reconstructing the input tokens, the input caption is reconstructed in a non-autoregressive

manner in the latent space of a Sentence-BERT (Reimers and Gurevych, 2019; Song et al., 2020) model. LTD can be implemented as an optimization constraint and as a dual-loss. Li et al. (2023b) show that contrastive losses are prone to feature suppression. They introduce predictive contrastive learning (PCL), which combines contrastive learning with a decoder to reconstruct the input data from the latent representations to prevent shortcut learning.

Adnan et al. (2022) measure the MI between the latent representation and the input as a domain agnostic metric to find where (and when) in training the neural network relies on shortcuts in the input data. Their main finding is that, in the presence of a shortcut, the MI between the input data and the latent representation of the data is lower than without a shortcut in the input data. Hence, the latent representation captures less information of the input data in the presence of the shortcut and mainly relies on the shortcut to predict the target.

6.7.4 Our focus

In this chapter, we focus on the problem of shortcut learning for VL in the context of multi-view VL representation learning with multiple captions per image. In contrast with previous (uni-modal) work on multi-view learning, we consider different captions matching to the same image as different *views*. We examine the problem by introducing a framework of synthetic shortcuts designed for VL representation learning, which allows us to investigate the problem in a controlled way. For our experiments, we select two prevalent VL models that are solely optimized with the InfoNCE loss: CLIP, a large-scale pre-trained model, and VSE++, a model trained from scratch. We select models that are solely optimized with a contrastive loss, to prevent measuring the effect of other optimization objectives on the shortcut learning problem.

6.8 DISCUSSION AND CONCLUSION

In this chapter, we focus on the shortcut learning problem of contrastive learning in the context of vision-language (VL) representation learning with multiple captions per image. We have proposed synthetic shortcuts for vision-language (SVL): a training and evaluation framework to examine the problem

of shortcut learning in a controlled way. The key component of this framework is synthetic shortcuts that we add to image-text data. Synthetic shortcuts represent additional, easily identifiable information that is shared between images and captions. We fine-tune CLIP and train a VSE++ model from scratch using our training framework to evaluate how prone contrastive VL models are to shortcut learning. Next, we have evaluated how shortcut learning can be partially mitigated using latent target decoding and implicit feature modification.

Main Findings. We have conducted experiments on two distinct VL models, CLIP and VSE++, and have evaluated the performance on Flickr30k and MS-COCO. We have found that when training with unique shortcuts, CLIP suppresses pre-trained features in favor of the shortcuts. VSE++ only learns to represent the shortcuts, when using unique shortcuts, showing that none of the remaining task-relevant (both shared and unique) information is captured by the encoders when training a model from scratch. When using *n bits of shortcuts*, we have shown that the more bits we use, the more the contrastive VL models rely on the synthetic shortcuts. Our results demonstrate that contrastive VL methods tend to depend on easy-to-learn discriminatory features shared among images and all matching captions while suppressing the remaining task-relevant information. Therefore, have answered the first part of the fifth research question of this thesis positively: we demonstrate that contrastive image-text methods predominantly rely on shortcuts when present in image-text training data. Next, we have evaluated two methods for reducing shortcut learning on our framework of synthetic shortcuts for image-caption datasets. Both methods partially mitigate shortcut learning when training and evaluating with our shortcut learning framework. Thus, we also answered the second part of the fifth research question of this thesis: we can reduce shortcuts in contrastive image-text representation learning, however, only to a certain extent. These findings show that our framework is a challenging framework to study and evaluate shortcut learning for contrastive VL and underline the complexity of our framework in studying and evaluating shortcut learning within the context of contrastive VL representation learning.

Implications. The implications of our findings are twofold. First, we examine the limitations of contrastive optimization objectives for VL representation learning, demonstrating that they predominantly capture features that are easily discriminable but may not necessarily constitute task-optimal representations. Second, this chapter contributes a novel framework for investigating the

shortcut learning problem in the context of VL representation learning with multiple captions per image, providing insights into the extent to which models rely on shortcuts when they are available and how existing shortcut reduction methods are capable of reducing shortcut learning when training with our framework.

Limitations. Some of the limitations of this chapter are related to the fact that we focused on two specific models, one optimization objective (InfoNCE), and two datasets, and the generalizability of our findings to other VL models, optimization objectives, and datasets warrants further exploration. Additionally, the synthetic shortcuts introduced in this chapter are not dependent on image-caption pairs. Our training and evaluation setup shows that, in the presence of shortcuts in the training data, contrastive VL models mainly rely on the easy-to-detect shortcut features, which indicates that the InfoNCE loss cannot learn tasks-optimal representations for VL tasks when multiple captions are used for training. However, it remains unclear to what degree the unique information of the captions is captured by the contrastive loss VL models.

Future Work. We suggest working on the development of optimization objectives that specifically address the shortcut learning problem for VL training with multiple captions per image. Moreover, we suggest extending our synthetic shortcuts for image-caption datasets to a framework with unique (shortcut) information per caption. By having unique shortcut information per caption, it becomes possible to measure how much of the shared/caption-specific shortcut information is captured by the encoder models. Another direction for future research includes investigating alternative training strategies or loss functions to further mitigate shortcut learning problems. More generally, we encourage the exploration of the generalizability of our findings across various VL models, different optimization functions (i.e., non-contrastive), and datasets.

Chapter Appendix

6.A NOTATION AND VARIABLES

Table 6.A.1: Overview of the notation and variables used throughout Chapter 6.

Symbol	Description
$\mathcal{L}_{\text{InfoNCE}}$	InfoNCE loss
$\mathcal{L}_{\text{InfoNCE+LTD}}$	Loss that combines InfoNCE and latent target decoding (LTD)
$\mathcal{L}_{\text{InfoNCE+IFM}}$	Loss that combines InfoNCE and implicit feature modification (IFM)
$\mathcal{D}$	Dataset $\mathcal{D}$ that comprises N image-caption tuples: $\mathcal{D} = \left\{ \left(\mathbf{x}_{\mathcal{I}}^{i}, \{\mathbf{x}_{\mathcal{C}_j}^{i}\}_{j=1}^{k} \right) \right\}_{i=1}^{N}$; i-th image-caption tuple in the dataset $\mathcal{D}$ consist out of an image $\mathbf{x}_{\mathcal{I}}^{i}$ and k associated captions $\{\mathbf{x}_{\mathcal{C}_j}^{i}\}_{j=1}^{k}$
$\mathcal{B}$	Batch of image-caption pairs
$\mathbf{x}_{\mathcal{I}}$	Image
$\mathbf{x}_{\mathcal{C}}$	Caption
$\mathbf{z}_{\mathcal{I}}$	Latent representation of image $\mathbf{x}_{\mathcal{I}}$
$\mathbf{z}_{\mathcal{C}}$	Latent representation of caption $\mathbf{x}_{\mathcal{C}}$
$\mathbf{z}_{\mathcal{C}\to\mathcal{I}}^{SUF}$	Latent representation of the caption $\mathbf{x}_{\mathcal{C}}$ that is sufficient for the image $\mathbf{x}_{\mathcal{I}}$
$\mathbf{z}_{\mathcal{I}\to\mathcal{C}}^{SUF}$	Latent representation of the image $\mathbf{x}_{\mathcal{I}}$ sufficient for the caption $\mathbf{x}_{\mathcal{C}}$
$\mathbf{z}_{\mathcal{I}\to\mathcal{C}}^{MIN}$	Latent representation of the image $\mathbf{x}_{\mathcal{I}}$ that is minimal sufficient for the caption $\mathbf{x}_{\mathcal{C}}$
$\mathbf{z}_{\mathcal{I}\to K}^{OPT}$	Latent representation of the image $\mathbf{x}_{\mathcal{I}}$ that is optimal for the set of captions K given the task T
R	Task-relevant information
$\neg R$	Task-irrelevant information
C	Task-relevant information specific for a caption $\mathbf{x}_{\mathcal{C}}$
S_{SynSC}	Synthetic shortcut
S	Original shared information
S^{+}	Shared information that includes synthetic shortcut
R^{+}	Task-relevant information that contains synthetic shortcut
$f_\theta(\cdot)$	Image encoder parametrised by θ; takes image $\mathbf{x}_{\mathcal{I}}$ as input and returns its latent representation $\mathbf{z}_{\mathcal{I}}$: $\mathbf{z}_{\mathcal{I}} := f_\theta(\mathbf{x}_{\mathcal{I}})$
$g_\phi(\cdot)$	Caption encoder parametrised by ϕ; takes caption $\mathbf{x}_{\mathcal{C}}$ as input and returns its latent representation $\mathbf{z}_{\mathcal{C}}$: $\mathbf{z}_{\mathcal{C}} := g_\phi(\mathbf{z}_{\mathcal{C}})$

6.B PROBLEM DEFINITION AND ASSUMPTIONS

In this chapter, we solely focus on contrastive VL representation learning. We work in a setting where we investigate the problem by fine-tuning a large pre-trained foundation model (CLIP, Radford et al., 2021) and training a resource-constrained image-text method from scratch (VSE++, Faghri et al., 2018). We train and evaluate using two benchmark datasets where multiple captions per image are available: Flickr30k (Young et al., 2014) and MS-COCO Captions (Lin et al., 2014). Both datasets come with 5 captions per image. We work in a dual-encoder setup, i.e., we have a separate image and caption encoder, which do not share parameters.

6.B.1 Evaluation task

The image-caption retrieval (ICR) evaluation task, consists of two sub-tasks: image-to-text (i2t) and text-to-image (t2i) retrieval. In ICR, either an image or a caption is used as a query and the goal is to rank a set of candidates in the other modality. In this work, we follow the standard ICR evaluation procedure (see, e.g., Faghri et al., 2018; Lee et al., 2018; Li et al., 2019a). The evaluation metric for the ICR task is recall@k, with $k = \{1, 5, 10\}$. For t2i retrieval, there is one matching/positive image per query caption (when using the Flickr30k or MS-COCO or dataset). Hence, the recall@k metric represents how often the correct image is present in the top-k of the ranking. For i2t retrieval, however, there are 5 matching captions per image. Therefore, only the highest-ranked correct caption is taken into account when measuring the recall@k (i.e., in the highest-ranked caption present in the top k). Standard practice to select the best model checkpoint during training is to use the *recall sum* (rsum) as a validation metric. The recall sum is the sum of recall at 1, 5, and 10, for both i2t and t2i. Therefore, the maximum value of the recall sum is 600.

6.B.2 Assumptions

Throughout this chapter, we rely on several assumptions about the problem definition. Our assumptions are defined at the level of an image-text tuple.

Following Section 6.2, we formalize the assumptions on the case where one image is associated with two captions: $\left(\mathbf{x}_{\mathcal{I}}, \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}\right)$.

Assumption 1. *Each caption in the tuple contain information that is distinct from the other captions in the tuple and all captions and image in the tuple contain shared and unique information:*

$$I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A}; \mathbf{x}_{\mathcal{C}_B}) > 0$$
$$I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A} \mid \mathbf{x}_{\mathcal{C}_B}) > 0,\ I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_B} \mid \mathbf{x}_{\mathcal{C}_A}) > 0.$$

Assumption 2. *Task-relevant information R is the combination of all the information shared between an image and each caption in the tuple:*

$$R = I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A} \mid \mathbf{x}_{\mathcal{C}_B}) + I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_B} \mid \mathbf{x}_{\mathcal{C}_A}) + I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A}; \mathbf{x}_{\mathcal{C}_B}).$$

6.C ANALYSIS OF CONTRASTIVE LEARNING FOR MULTIPLE CAPTIONS PER IMAGE

Theorem 6.1 (Suboptimality of contrastive learning with multiple captions per image)**.** *Given an image $\mathbf{x}_{\mathcal{I}}$, a set of matching captions $\mathcal{C} = \{\mathbf{x}_{\mathcal{C}_A}, \mathbf{x}_{\mathcal{C}_B}\}$, and a contrastive learning loss function $\mathcal{L}_{InfoNCE}$ that optimizes for task T, image representations learned during contrastive learning will be minimal sufficient and will never be task-optimal image representations. More formally, assume that:*

(H_1) *$\forall i, j \in \{A, B\}$ such that $i \neq j$, $I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_i}; \mathbf{x}_{\mathcal{C}_i}) = I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_i} \mid \mathbf{x}_{\mathcal{C}_j}) + I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_i}; \mathbf{x}_{\mathcal{C}_j})$.*

(H_2) *$\exists i, j \in \{A, B\}$ with $i \neq j$ such that $I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_i} \mid \mathbf{x}_{\mathcal{C}_j}) > 0$.*

Then the following holds:

(T_2) *$\exists\, i \in \{A, B\}$ such that $I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B}) > I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_i}; \mathbf{x}_{\mathcal{C}_i})$.*

Proof. Following Eq. 6.1 we define a task-optimal representation of an image $\mathbf{x}_{\mathcal{I}}$ w.r.t. all matching captions in $\mathcal{C}$ as:

$$I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}}; \mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B}) = \underbrace{I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A} \mid \mathbf{x}_{\mathcal{C}_B})}_{C_A} + \underbrace{I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_B} \mid \mathbf{x}_{\mathcal{C}_A})}_{C_B} + \underbrace{I(\mathbf{x}_{\mathcal{I}}; \mathbf{x}_{\mathcal{C}_A}; \mathbf{x}_{\mathcal{C}_B})}_{S}.$$

Furthermore, following Definition 6.2.3, we define minimal sufficient representations of image $\mathbf{x}_{\mathcal{I}}$ w.r.t. each matching caption in $\mathcal{C}$ as a combination of caption-specific and shared information:

$$I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_A}) = \underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A}\mid\mathbf{x}_{\mathcal{C}_B})}_{C_A} + \underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{S}$$
$$I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_B};\mathbf{x}_{\mathcal{C}_B}) = \underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_B}\mid\mathbf{x}_{\mathcal{C}_A})}_{C_B} + \underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{S}.$$

Following assumption H_2, for at least one caption $\mathbf{x}_{\mathcal{C}} \in \mathcal{C}$ associated with the image $\mathbf{x}_{\mathcal{I}}$, caption-specific information is positive. Therefore, we consider two cases:

- If caption-specific information of $\mathbf{x}_{\mathcal{C}_A}$ is positive, that is, if $I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A}\mid\mathbf{x}_{\mathcal{C}_B}) > 0$:

$$\underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A}\mid\mathbf{x}_{\mathcal{C}_B}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_B}\mid\mathbf{x}_{\mathcal{C}_A}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}};\mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B})} >$$
$$\underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_B}\mid\mathbf{x}_{\mathcal{C}_A}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_B};\mathbf{x}_{\mathcal{C}_B})} \Rightarrow$$
$$\Rightarrow I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}};\mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B}) > I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_B};\mathbf{x}_{\mathcal{C}_B}).$$

- Similarly, if caption-specific information of $\mathbf{x}_{\mathcal{C}_B}$ is positive, that is, if $I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_B}\mid\mathbf{x}_{\mathcal{C}_A}) > 0$:

$$\underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A}\mid\mathbf{x}_{\mathcal{C}_B}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_B}\mid\mathbf{x}_{\mathcal{C}_A}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}};\mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B})} >$$
$$\underbrace{I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A}\mid\mathbf{x}_{\mathcal{C}_B}) + I(\mathbf{x}_{\mathcal{I}};\mathbf{x}_{\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_B})}_{I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_A})} \Rightarrow$$
$$\Rightarrow I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}};\mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B}) > I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_A};\mathbf{x}_{\mathcal{C}_A}).$$

Therefore, we show that in a setup where a single image is associated with multiple captions, and given at least one caption contains caption-specific information, image representations learned contrastively w.r.t. associated captions would contain less information than task-optimal image representation: $\exists\, i \in \{A, B\}$ such that $I(\mathbf{z}^{OPT}_{\mathcal{I}\to\mathcal{C}};\mathbf{x}_{\mathcal{C}_A}\mathbf{x}_{\mathcal{C}_B}) > I(\mathbf{z}^{MIN}_{\mathcal{I}\to\mathcal{C}_i};\mathbf{x}_{\mathcal{C}_i})$. □

6.D EXPERIMENTAL SETUP

6.D.1 Datasets

Flickr30k consists of 31 000 images annotated with 5 matching captions (Young et al., 2014).

MS-COCO consists of 123 287 images, each image annotated with 5 matching captions (Lin et al., 2014). The original dataset was introduced for large-scale object recognition.

For both datasets, we use the training, validation, and test splits from (Karpathy and Li, 2015).

6.D.2 Models

We use CLIP and VSE++. Both consist of an image and a text encoder that do not share parameters.

CLIP is a large-scale image-text foundation model (Radford et al., 2021). The model is pre-trained on a collection of 400 million image-text pairs collected from the Web. The encoders are pre-trained using a contrastive loss (InfoNCE) on image-text pairs. The text encoder of consists of a 12-layer transformer model, described in (Radford et al., 2019). As for the image encoder, CLIP utilizes various model backbones, such as ResNet (He et al., 2016) and vision transformer (Dosovitskiy et al., 2021). In this work, we use the ResNet-50 ('RN50') variant of the CLIP image encoder.[3] The CLIP encoders are trained to jointly understand images and text. Therefore, the learned representations generalize to a wide range of different zero-shot (visual) evaluation tasks, such as classification, without task-specific fine-tuning, by using textual prompts.

VSE++ is an image-caption encoder trained from scratch (Faghri et al., 2018). The model features a triplet loss function with a margin parameter $\alpha = 0.2$. The text encoder is a one-layer gated recurrent unit (GRU) (Cho et al., 2014b). The available image encoder configurations are ResNet-152 (He et al., 2016) and VGG19 (Simonyan and Zisserman, 2015). In this work, we use ResNet-152.

3 https://github.com/openai/CLIP/

6.D.3 Training

CLIP. To fine-tune CLIP, we follow (Yuksekgonul et al., 2023). All models are fine-tuned for 5 epochs. We employ a cosine-annealing learning rate schedule, with a base learning rate of $2e-5$, and 100 steps of warm-up. As an optimizer, we use AdamW (Loshchilov and Hutter, 2019) with a gradient clipping value of 2. For the InfoNCE loss, we use the logit-scale (i.e., temperature τ) from the pre-trained CLIP model and fine-tune the logit-scale end-to-end along with the rest of the model parameters.

VSE++. The model is trained for 30 epochs using a linear learning rate schedule with a base learning rate of $2e-4$. We use the Adam optimizer (Kingma and Ba, 2015) with a gradient clipping value of 2. Instead of the triplet loss, we use the InfoNCE loss similar to (Radford et al., 2021),

For both models, instead of selecting the best-performing model based on the validation set scores, we use the final checkpoint at the end of training.

6.D.4 Shortcut sampling

Our goal is to add the shortcuts in a manner that preserves the original information of the images and captions. For the captions, we append the shortcut at the end of the captions. In order to prevent a tokenizer from tokenizing the shortcut into a single token, we insert spaces between each number of the shortcut. For the images, we place the numbers of the shortcuts at the top of the images, evenly spaced across the entire width of the images (to make sure the shortcut is evenly spaced across the feature map of the image). We always use 6 digits to represent a shortcut. If a shortcut number contains fewer than 6 digits, we fill the remaining positions with zeros for padding. For the MNIST images, we always sample a random image from the set of images representing the number that belongs to (also during evaluation), to prevent overfitting on specific MNIST images. In Figure 6.D.1, we provide four examples of image-caption pairs with randomly added shortcuts. The examples in Figure 6.D.1 show (i) how synthetic shortcuts are added to the image and the caption, and (ii) that the shortcuts preserve the original (task-relevant) information of the images and captions.

(a) Caption: "A bathroom sink with wood finish cabinets. 0 3 9 9 6 5."

(b) Caption: "A guy in a brown shirt has just hit a tennis ball. 0 7 7 1 1 4."

(c) Caption: "A man in shorts is lying on the beach. 0 0 6 9 9 3."

(d) Caption: "A player up to bat in a baseball game. 1 0 1 9 9 2."

Figure 6.D.1: Four random samples from the MS-COCO dataset including shortcuts added on both the image and caption.

7 Conclusion

In this thesis, we have presented five research questions, centered around the topic of multi-modal learning problems and algorithms. Throughout the thesis, we have focused on three modalities: (i) *audio*, (ii) *image(s)*, and (iii) *text*. The investigation of these three modalities has been centered around three multi-modal tasks: (i) *automatic speech recognition*, (ii) *scene text recognition*, and (iii) *image-caption retrieval* (or more broadly image-text representation learning). We have divided this thesis into two parts. In Part 1, the main focus has been on multi-modal *sequence modeling* (Chapter 2 and Chapter 3). For each sequence modeling task, we have studied in this thesis (automatic speech recognition (ASR) and scene text recognition (STR)), we have introduced a novel method: a hard negative mining approach for contextual ASR and a unified network architecture for bidirectional STR. In Part 2, we have shifted our focus to multi-modal *representation learning* for images and text (Chapter 4, Chapter 5, and Chapter 6). We have investigated contrastive image-text representation learning, where we have provided new insights into the understanding and improvement of contrastive image-text methods.

In the final chapter of this thesis, we first summarize the research questions and main findings outlined in each research chapter (Section 7.1). We conclude with directions for future work (Section 7.2).

7.1 SUMMARY OF FINDINGS

Part 1: Multi-modal sequence modeling

Research Question 1: *Can we improve contextual automatic speech recognition by introducing an efficient online hard negative phrase mining approach?*

In Chapter 2, which is based on (Bleeker et al., 2023a), we have answered the first research question of this thesis positively by introducing *approximate nearest neighbour phrase* (ANN-P) *mining* for contextual ASR. ANN-P mining is an efficient, online hard negative mining approach that can be combined with a context-aware transformer transducer. We show that using ANN-P mining results in up to 7% relative word error rate reductions for the personalized portion of the test data in streaming scenarios.

Research Question 2: *Can we unify bidirectional multi-modal sequence modeling into a single decoder architecture for scene text recognition?*

In Chapter 3, which is based on (Bleeker and de Rijke, 2020), we have answered the second research question of this thesis positively by introducing the *bidirectional scene text transformer* (Bi-STET). Bi-STET is a novel transformer-based bidirectional STR method with a single decoder for both decoding directions. Bi-STET outperforms the bidirectional STR method by (Shi et al., 2018), which uses two decoders, and performs on par with or outperforms other state-of-the-art STR methods. Moreover, Bi-STET shows its strength in handling curved and rotated text without specific rectification components.

Part 2: Image-text representation learning

Research Question 3: *Do lessons from metric learning generalize to image-caption retrieval?*

In Chapter 4, which is based on (Bleeker and de Rijke, 2022), we have critically examined a diverse set of metric learning functions and investigated if the findings from metric learning generalize to the image-caption retrieval (ICR) task. We have answered the third research question of this thesis negatively: the lessons from metric learning do not generalize to the ICR task. To gain a better understanding of why a certain loss function performs better than others, we have introduced the *counting contributing samples* (COCOS) method. The COCOS method shows us that, on average, the highest performing loss function takes at most one negative sample into account when computing the gradient. This is in contrast to the underperforming contrastive losses that take too many (non-informative) negative samples into account in the gradient computation.

Research Question 4: *Can we reduce predictive feature suppression for resource-constrained contrastive image-text representation learning?*

In Chapter 5, which is based on (Bleeker et al., 2023b), we have investigated the problem of predictive feature suppression for resource-constrained ICR methods. We have answered the fourth research question of this thesis positively by introducing *latent target decoding* (LTD). LTD is a reconstruction objective, which can be combined with a contrastive loss, that reconstructs the input caption in a non-auto-regressive manner in the latent space of a general-purpose sentence encoder (as opposed to reconstructing the input tokens). We show that constraint-based LTD outperforms ICR methods that are solely trained with a contrastive loss and that implementing LTD as an optimization constraint is more effective than a dual objective. Moreover, we show that LTD can be applied with different contrastive losses and ICR methods, offering novel solutions to reduce predictive feature suppression for resource-constrained image-text representation learning.

Research Question 5: *Can we demonstrate and reduce shortcuts in contrastive image-text representation learning?*

In Chapter 6, which is based on (Bleeker et al., 2024), we have taken another look at image-text representation learning by investigating the shortcut learning problem. To answer this thesis's fifth and final research question, we introduced the *synthetic shortcuts for vision-language* (SVL) framework. The SVL framework is a training and evaluation framework, that allows us to injecting synthetic shortcuts into image-text data. By injecting synthetic shortcuts in a controllable manner into the training and evaluation data, it becomes possible to measure (and therefore demonstrate) to what extent contrastive image-text methods depend on a shortcut in the training data when minimizing the contrastive objective. We find that contrastive image-text methods that are either trained from scratch or fine-tuned with data containing these synthetic shortcuts mainly learn to represent the shortcut features while suppressing the remaining task-relevant information in the input data. Therefore, we conclude that contrastive losses are insufficient to learn task-optimal image-text representations (i.e., contain all relevant information w.r.t. an evaluation task). Finally, we examined two shortcut reduction methods on the SVL framework. We find that shortcut solutions can partially be mitigated by using the short-

cut reduction methods in some settings when training and evaluating with the SVL framework.

7.2 FUTURE WORK

In the concluding section of each research chapter in this thesis, we have presented recommendations for future work (except for Chapter 2). In this section, we take a step back and provide recommendations for future work from a broader perspective. We focus on two broad directions: non-auto-regressive visual-language models and inductive biases for efficient multi-modal representation learning.

7.2.1 Non-auto-regressive visual-language models for multi-modal representation learning

Throughout this thesis, we mainly focused on contrastive image-text methods for multi-modal representation learning. Since the introduction of large language models (e.g., Radford et al., 2019; Brown et al., 2020b; Chowdhery et al., 2023), visual-language models (VLMs) (e.g., Alayrac et al., 2022; Li et al., 2022a; Wang et al., 2022b; Li et al., 2023a) have become popular as well. Most VLMs use (frozen) pre-trained contrastive vision encoders as their backbone for visual representations (Tong et al., 2024) (such as CLIP (Radford et al., 2019)). However, solely using a contrastive optimization objective results in visual (and textual) representations that seem to lack all kinds of linguistic properties (Yuksekgonul et al., 2023; Tong et al., 2024) or represent a shortcut (Chapter 6). Therefore, it might be possible that the strong performance of VLMs primarily depends on the quality of the language decoder, and not necessarily on the visual representations. As long as a text decoder of VLMs is conditioned on the previously predicted tokens to generate the output sequence, it remains unclear to what extent textual and linguistic information is captured by the visual representations. By using non-auto-regressive text generation objectives during training, the vision encoder becomes a bottleneck that is forced to capture all the linguistic information and understanding that is needed to generate the output sequence (since the output does not depend on the previously predicted tokens). In Chapter 5, we have started to explore

non-auto-regressive textual reconstruction by introducing latent target decoding. Furthermore, Tschannen et al. (2023) have taken the first step in large-scale settings, by introducing a non-auto-regressive caption reconstruction objective for transformer-based VLMs models trained from scratch. Nevertheless, we are convinced that there is still significant potential to efficiently improve the performance of the visual and textual representations and therefore we advocate for the exploration of new non-auto-regressive training objectives in the VLMs domain.

7.2.2 Multi-modal specific inductive biases for efficient representation learning

Many of the same representation learning principles are applied among different modalities. For instance, InfoNCE-based losses serve as a prominent choice of optimization function for audio (e.g., van den Oord et al., 2018; Baevski et al., 2020), images (e.g., Chen et al., 2020c), image–text (e.g., Jia et al., 2021; Radford et al., 2021), and text (e.g., Reimers and Gurevych, 2019) representation learning. However, in Chapter 6 and (Bleeker et al., 2024), we demonstrate that the InfoNCE loss may not be ideal for capturing the one-to-many relationship between images and text. The success of joint-embedding image-text encoder models, such as ALIGN (Jia et al., 2021), CLIP (Radford et al., 2021), and Florence (Yuan et al., 2021), is predominantly attributed to the scale of the training setup rather than modeling task and data inductive biases in either the model or the optimization objective. As a result, to improve the performance of joint-embedding image-text models without relying on the scale of the training setup, we advocate for the exploration of data and compute-efficient representation learning methods that leverage data-specific inductive biases. The overall objectives would be: (i) to improve the quality of the learned representations with a focus on specific applications, and (ii) to increase the data and computational efficiency of the representation learning methods (considering computational costs).

Bibliography

Mohammed Adnan, Yani A. Ioannou, Chuan-Yung Tsai, Angus Galloway, Hamid R. Tizhoosh, and Graham W. Taylor (2022). "Monitoring Shortcut Learning using Mutual Information." In: *arXiv preprint arXiv:2206.13034*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan (2022). "Flamingo: a Visual Language Model for Few-Shot Learning." In: *NeurIPS*, pp. 23716–23736.

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter (2019). "Fusion of Detected Objects in Text for Visual Question Answering." In: *EMNLP*, pp. 2131–2140.

Uri Alon, Golan Pundak, and Tara N. Sainath (2019). "Contextual Speech Recognition with Difficult Negative Training Examples." In: *ICASSP*, pp. 6440–6444.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould (2016). "SPICE: Semantic Propositional Image Caption Evaluation." In: *ECCV*, pp. 382–398.

Philip Bachman, R. Devon Hjelm, and William Buchwalter (2019). "Learning Representations by Maximizing Mutual Information Across Views." In: *NeurIPS*, pp. 15509–15519.

Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee (2019). "What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis." In: *ICCV*, pp. 4714–4722.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli (2020). "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations." In: *NeurIPS*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate." In: *ICLR*.

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency (2019). "Multimodal Machine Learning: A Survey and Taxonomy." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, pp. 423–443.

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei (2022). "VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts." In: *NeurIPS*, pp. 32897–32912.

Adrien Bardes, Jean Ponce, and Yann LeCun (2022). "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning." In: *ICLR*.

Lawrence W. Barsalou (2001). "The Human Conceptual System." In: *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, p. 186. ISBN: 1581133774.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent (2013). "Representation Learning: A Review and New Perspectives." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, pp. 1798–1828.

Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven (2013). "PhotoOCR: Reading Text in Uncontrolled Conditions." In: *ICCV*, pp. 785–792.

Ali Furkan Biten, Andrés Mafla, Lluís Gómez, and Dimosthenis Karatzas (2022). "Is An Image Worth Five Sentences? A New Look into Semantics for Image-Text Matching." In: *WACV*, pp. 2483–2492.

Maurits Bleeker and Maarten de Rijke (2020). "Bidirectional Scene Text Recognition with a Single Decoder." In: *ECAI*, pp. 2664–2672.

Maurits Bleeker and Maarten de Rijke (2022). "Do Lessons from Metric Learning Generalize to Image-Caption Retrieval?" In: *ECIR*, pp. 535–551.

Maurits Bleeker, Mariya Hendriksen, Andrew Yates, and Maarten de Rijke (2024). "Demonstrating and Reducing Shortcuts in Vision-Language Representation Learning." In: *arXiv preprint arXiv:2402.17510*.

Maurits Bleeker, Pawel Swietojanski, Stefan Braun, and Xiaodan Zhuang (2023a). "Approximate Nearest Neighbour Phrase Mining for Contextual Speech Recognition." In: *Interspeech*.

Maurits Bleeker, Andrew Yates, and Maarten de Rijke (2023b). "Reducing Predictive Feature Suppression in Resource-Constrained Contrastive Image-Caption Retrieval." In: *Transactions on Machine Learning Research*.

Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman (2020a). "Smooth-AP: Smoothing the Path Towards Large-Scale Image Retrieval." In: *ECCV*, pp. 677–694.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020b). "Language Models are Few-Shot Learners." In: *NeurIPS*, pp. 1877–1901.

Sebastian Bruch, Xuanhui Wang, Michael Bendersky, and Marc Najork (2019a). "An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance." In: *ICTIR*, pp. 75–78.

Sebastian Bruch, Masrour Zoghi, Michael Bendersky, and Marc Najork (2019b). "Revisiting Approximate Metric Optimization in the Age of Deep Neural Networks." In: *SIGIR*, pp. 1241–1244.

Antoine Bruguier, Rohit Prabhavalkar, Golan Pundak, and Tara N. Sainath (2019). "Phoebe: Pronunciation-aware Contextualization for End-to-End Speech Recognition." In: *ICASSP*, pp. 6171–6175.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals (2015). "Listen, Attend and Spell." In: *arXiv preprint arXiv:1508.01211*.

Feng-Ju Chang, Jing Liu, Martin Radfar, Athanasios Mouchtaris, Maurizio Omologo, Ariya Rastrow, and Siegfried Kunzmann (2021). "Context-Aware Transformer Transducer for Speech Recognition." In: *Automatic Speech Recognition and Understanding Workshop*, pp. 503–510.

Hui Chen, Guiguang Ding, Zijia Lin, Sicheng Zhao, and Jungong Han (2019a). "Cross-Modal Image-Text Retrieval with Semantic Consistency." In: *ACM MM*, pp. 1749–1757.

Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han (2020a). "IMRAM: Iterative Matching With Recurrent Attention Memory for Cross-Modal Image-Text Retrieval." In: *CVPR*, pp. 12652–12660.

Tianlang Chen, Jiajun Deng, and Jiebo Luo (2020b). "Adaptive Offline Quintuplet Loss for Image-Text Matching." In: *ECCV*, pp. 549–565.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton (2020c). "A Simple Framework for Contrastive Learning of Visual Representations." In: *ICML*, pp. 1597–1607.

Ting Chen, Calvin Luo, and Lala Li (2021a). "Intriguing Properties of Contrastive Losses." In: *NeurIPS*, pp. 11834–11845.

Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li (2021b). "Developing Real-time Streaming Transformer Transducer for Speech Recognition on Large-scale Dataset." In: *ICASSP*, pp. 5904–5908.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick (2015). "Microsoft COCO Captions: Data collection and evaluation server." In: *arXiv preprint arXiv:1504.00325*.

Xinlei Chen and Kaiming He (2021). "Exploring Simple Siamese Representation Learning." In: *CVPR*, pp. 15750–15758.

Xuanang Chen, Ben He, Kai Hui, Le Sun, and Yingfei Sun (2021c). "Simplified TinyBERT: Knowledge Distillation for Document Retrieval." In: *ECIR*, pp. 241–248.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020d). "UNITER: UNiversal Image-TExt Representation Learning." In: *ECCV*, pp. 104–120.

Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L. Seltzer, and Christian Fuegen (2019b). "Joint Grapheme and Phoneme Embeddings for Contextual End-to-End ASR." In: *Interspeech*, pp. 3490–3494.

Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou (2017). "Focusing Attention: Towards Accurate Text Recognition in Natural Images." In: *ICCV*, pp. 5076–5084.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio (2014a). "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches." In: *EMNLP Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014b). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." In: *ACL*, pp. 1724–1734.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel

(2023). "PaLM: Scaling Language Modeling with Pathways." In: *Journal of Machine Learning Research* 24, 240:1–240:113.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus (2021). "Probabilistic Embeddings for Cross-Modal Retrieval." In: *CVPR*, pp. 8415–8424.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "ImageNet: A large-scale Hierarchical Image Database." In: *CVPR*, pp. 248–255.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *NAACL-HLT*.

Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu (2021). "Similarity Reasoning and Filtration for Image-Text Matching." In: *AAAI*, pp. 1218–1226.

Saket Dingliwal, Monica Sunkara, Srikanth Ronanki, Jeff Farris, Katrin Kirchhoff, and Sravan Bodapati (2023). "Personalization of CTC Speech Recognition Models." In: *Spoken Language Technology Workshop*, pp. 302–309.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In: *ICLR*.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler (2018). "VSE++: Improving Visual-Semantic Embeddings with Hard Negatives." In: *BVCM*, p. 12.

Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata (2020). "Learning Robust Representations via Multi-View Information Bottleneck." In: *ICLR*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks." In: *ICML*, pp. 1126–1135.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant (2021). "SPLADE: Sparse lexical and expansion model for first stage ranking." In: *SIGIR*, pp. 2288–2292.

Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov (2013). "DeViSE: A Deep Visual-Semantic Embedding Model." In: *NeurIPS*, pp. 2121–2129.

Luyu Gao, Zhuyun Dai, and Jamie Callan (2021). "Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline." In: *ECIR*, pp. 280–286.

Yunze Gao, Yingying Chen, Jinqiao Wang, and Hanqing Lu (2017). "Reading Scene Text with Attention Convolutional Sequence Modeling." In: *arXiv preprint arXiv:1709.04303*.

Sushant Gautam (2023). "Bridging Multimedia Modalities: Enhanced Multimodal AI Understanding and Intelligent Agents." In: *ICMI*, pp. 695–699.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann (2020). "Shortcut Learning in Deep Neural Networks." In: *Nature Machine Intelligence* 11, pp. 665–673.

Xavier Glorot and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks." In: *AISTATS*, pp. 249–256.

Alex Graves (2012). "Sequence Transduction with Recurrent Neural Networks." In: *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber (2006). "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks." In: *ICML*, pp. 369–376.

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber (2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition." In: *Transactions on Pattern Analysis and Machine Intelligence* 31, pp. 855–868.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko (2020). "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning." In: *NeurIPS*, pp. 21271–21284.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang (2020). "Conformer: Convolution-augmented Transformer for Speech Recognition." In: *Interspeech*, pp. 5036–5040.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft (2016). "A Deep Relevance Matching Model for Ad-hoc Retrieval." In: *CIKM*, pp. 55–64.

Jinxi Guo, Tara N. Sainath, and Ron J. Weiss (2019a). "A Spelling Correction Model for end-to-end Speech Recognition." In: *ICASSP*, pp. 5651–5655.

Wenzhong Guo, Jianwen Wang, and Shiping Wang (2019b). "Deep Multimodal Representation Learning: A Survey." In: *IEEE Access* 7, pp. 63373–63394.

Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman (2016). "Synthetic Data for Text Localisation in Natural Images." In: *CVPR*, pp. 2315–2324.

Raia Hadsell, Sumit Chopra, and Yann LeCun (2006). "Dimensionality Reduction by Learning an Invariant Mapping." In: *CVPR*, pp. 1735–1742.

Keith B. Hall, Eunjoon Cho, Cyril Allauzen, Françoise Beaufays, Noah Coccaro, Kaisuke Nakajima, Michael Riley, Brian Roark, David Rybach, and Linda Zhang (2015). "Composition-based On-the-Fly Rescoring for Salient N-gram Biasing." In: *Interspeech*, pp. 1418–1422.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. (2014). "Deep speech: Scaling up end-to-end speech recognition." In: *arXiv preprint arXiv:1412.5567*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick (2020). "Momentum Contrast for Unsupervised Visual Representation Learning." In: *CVPR*, pp. 9726–9735.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition." In: *CVPR*, pp. 770–778.

Katherine L. Hermann and Andrew K. Lampinen (2020). "What shapes feature representations? Exploring datasets, architectures, and training." In: *NeurIPS*, pp. 9995–10006.

Geoffrey E. Hinton and Ruslan R Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks." In: *Science* 313, pp. 504–507.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean (2015). "Distilling the Knowledge in a Neural Network." In: *arXiv preprint arXiv:1503.02531*.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio (2019). "Learning Deep Representations by Mutual Information Estimation and Maximization." In: *ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber (1997). "Long Short-term Memory." In: *Neural Computation*, pp. 1735–1780.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson (2018). "Averaging Weights Leads to Wider Optima and Better Generalization." In: *UAI*, pp. 876–885.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman (2014a). "Deep Structured Output Learning for Unconstrained Text Recognition." In: *ICLR*.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman (2016). "Reading Text in the Wild with Convolutional Neural Networks." In: *International Journal of Computer Vision* 116, pp. 1–20.

Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman (2014b). "Deep Features for Text Spotting." In: *ECCV*, pp. 512–528.

Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf (2020). "Contextual RNN-T for Open Domain ASR." In: *Interspeech*, pp. 11–15.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig (2021). "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision." In: *ICML*, pp. 4904–4916.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian (2021). "Understanding Dimensional Collapse in Contrastive Self-supervised Learning." In: *ICLR*.

Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit (2017). "One Model To Learn Them All." In: *arXiv preprint arXiv:1706.05137*.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukás Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny (2015). "ICDAR 2015 Competition on Robust Reading." In: *ICDAR*, pp. 1156–1160.

Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernández Mota, Jon Almazán, and Lluis-Pere de las Heras (2013). "ICDAR 2013 Robust Reading Competition." In: *ICDAR*, pp. 1484–1493.

Andrej Karpathy and Fei-Fei Li (2015). "Deep Visual-Semantic Alignments for Generating Image Descriptions." In: *CVPR*, pp. 3128–3137.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih (2020). "Dense Passage Retrieval for Open-Domain Question Answering." In: *EMNLP*, pp. 6769–6781.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky (2018). "Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context." In: *ACL*, pp. 284–294.

Omar Khattab and Matei Zaharia (2020). "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT." In: *SIGIR*, pp. 39–48.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan (2020). "Supervised Contrastive Learning." In: *NeurIPS*, pp. 18661–18673.

Wonjae Kim, Bokyung Son, and Ildoo Kim (2021). "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision." In: *ICML*, pp. 5583–5594.

Diederik P. Kingma and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization." In: *ICLR*.

Thomas N. Kipf and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks." In: *ICLR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel (2014). "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models." In: *arXiv preprint arXiv:1411.2539*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2016). "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." In: *arXiv preprint arXiv:1602.07332*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In: *NeurIPS*, pp. 1106–1114.

Duc Le, Gil Keren, Julian Chan, Jay Mahadeokar, Christian Fuegen, and Michael L. Seltzer (2021). "Deep Shallow Fusion for RNN-T Personalization." In: *Spoken Language Technology Workshop*, pp. 251–257.

Chen-Yu Lee and Simon Osindero (2016). "Recursive Recurrent Nets with Attention Modeling for OCR in the Wild." In: *CVPR*, pp. 2231–2239.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John F. Canny, and Ian Fischer (2021). "Compressive Visual Representations." In: *NeurIPS*, pp. 19538–19552.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He (2018). "Stacked Cross Attention for Image-Text Matching." In: *ECCV*, pp. 212–228.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang (2020a). "Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training." In: *AAAI*, pp. 11336–11344.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi (2023a). "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models." In: *ICML*, pp. 19730–19742.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi (2022a). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation." In: *ICML*, pp. 12888–12900.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi (2021). "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation." In: *NeurIPS*, pp. 9694–9705.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu (2019a). "Visual Semantic Reasoning for Image-Text Matching." In: *ICCV*, pp. 4653–4661.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang (2019b). "VisualBERT: A Simple and Performant Baseline for Vision and Language." In: *arXiv preprint arXiv:1908.03557*.

Tianhong Li, Lijie Fan, Yuan Yuan, Hao He, Yonglong Tian, Rogerio Feris, Piotr Indyk, and Dina Katabi (2020b). "Making Contrastive Learning Robust to Shortcuts." In: *arXiv preprint arXiv:2012.09962*.

Tianhong Li, Lijie Fan, Yuan Yuan, Hao He, Yonglong Tian, Rogério Feris, Piotr Indyk, and Dina Katabi (2023b). "Addressing Feature Suppression in Unsupervised Visual Representations." In: *WACV*, pp. 1411–1420.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao (2020c). "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks." In: *ECCV*, pp. 121–137.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan (2022b). "Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm." In: *ICLR*.

Paul Pu Liang, Zihao Deng, Martin Q. Ma, James Zou, Louis-Philippe Morency, and Russ Salakhutdinov (2023). "Factorized Contrastive Learning: Going Beyond Multi-view Redundancy." In: *NeurIPS*.

Chin-Yew Lin (2004). "ROUGE: A Package for Automatic Evaluation of Summaries." In: *Text summarization branches out*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context." In: *ECCV*, pp. 740–755.

Erik Lindgren, Sashank J. Reddi, Ruiqi Guo, and Sanjiv Kumar (2021). "Efficient Training of Retrieval Models using Negative Cache." In: *NeurIPS*. Vol. 4134–4146.

Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang (2020). "Graph Structured Network for Image-Text Matching." In: *CVPR*, pp. 10918–10927.

Wei Liu, Chaofeng Chen, Kwan-Yee K. Wong, Zhizhong Su, and Junyu Han (2016). "STAR-Net: A SpaTial Attention Residue Network for Scene Text Recognition." In: *BMVC*, p. 7.

Ilya Loshchilov and Frank Hutter (2019). "Decoupled Weight Decay Regularization." In: *ICLR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks." In: *NeurIPS*, pp. 13–23.

Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Ya Liu, and Arnold Overwijk (2021a). "Less is More: Pre-training a Strong Siamese Encoder Using a Weak Decoder." In: *EMNLP*, pp. 2780–2791.

Xiaopeng Lu, Tiancheng Zhao, and Kyusong Lee (2021b). "VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words." In: *ACL/IJCNLP*, pp. 5020–5029.

Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young (2003). "ICDAR 2003 Robust Reading Competitions." In: *ICDAR*, pp. 682–687.

Thang Luong, Hieu Pham, and Christopher D. Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation." In: *EMNLP*, pp. 1412–1421.

Itzik Malkiel and Lior Wolf (2021). "MTAdam: Automatic Balancing of Multiple Training Loss Terms." In: *EMNLP*, pp. 10713–10729.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge University Press. ISBN: 978-0-521-86571-5.

Erik McDermott, Hasim Sak, and Ehsan Variani (2021). "A Density Ratio Approach to Language Model Fusion in End-To-End Automatic Speech Recognition." In: *Automatic Speech Recognition and Understanding Workshop*, pp. 434–441.

Zhong Meng, Naoyuki Kanda, Yashesh Gaur, Sarangarajan Parthasarathy, Eric Sun, Liang Lu, Xie Chen, Jinyu Li, and Yifan Gong (2021). "Internal Language Model Estimation for Domain-adaptive End-to-End Speech Recognition." In: *ICASSP*, pp. 7338–7342.

Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet (2020a). "Fine-Grained Visual Textual Alignment for Cross-Modal Retrieval Using Transformer Encoders." In: *ACM Transactions on Multimedua Computing, Communications, and Applications*, 128:1–128:23.

Nicola Messina, Fabrizio Falchi, Andrea Esuli, and Giuseppe Amato (2020b). "Transformer Reasoning Network for Image-Text Matching and Retrieval." In: *ICPR*, pp. 5222–5229.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space." In: *ICLR*.

Anand Mishra, Karteek Alahari, and C. V. Jawahar (2012). "Scene Text Recognition using Higher Order Language Priors." In: *BMVC*, pp. 1–11.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh (2017). "No Fuss Distance Metric Learning Using Proxies." In: *ICCV*, pp. 360–368.

Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie (2022). "SLIP: Self-supervision Meets Language-Image Pre-training." In: *ECCV*, pp. 529–544.

Tsendsuren Munkhdalai, Khe Chai Sim, Angad Chandorkar, Fan Gao, Mason Chua, Trevor Strohman, and Françoise Beaufays (2022). "Fast Contextual Adaptation with Neural Associative Memory for On-Device Personalized Speech Recognition." In: *ICASSP*, pp. 6632–6636.

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim (2020). "A Metric Learning Reality Check." In: *ECCV*, pp. 681–699.

Michael Neely, Stefan F. Schouten, Maurits Bleeker, and Ana Lucic (2021). "Order in the Court: Explainable AI Methods Prone to Disagreement." In: *Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*.

Thien Nguyen, Nathalie Tran, Liuhui Deng, Thiago Fraga da Silva, Matthew Radzihovsky, Roger Hsiao, Henry Mason, Stefan Braun, Erik McDermott, Dogan Can, Pawel Swietojanski, Lyan Verwimp, Sibel Oyman, Tresi Arvizo, Honza Silovsky, Arnab Ghoshal, Mathieu Martel, Bharat Ram Ambati, and Mohamed Ali (2022). "Optimizing Bilingual Neural Transducer with Synthetic Code-switching Text Generation." In: *arXiv preprint arXiv:2210.12214*.

Jan Noyes, Kate Garland, and D Bruneasu (2004). "Humans: Skills, capabilities and limitations." In: *Human Factors for Engineers; Sandorn, C., Harvey, RS, Eds*, pp. 35–56.

Harrie Oosterhuis and Maarten de Rijke (2018). "Differentiable Unbiased Online Learning to Rank." In: *CIKM*, pp. 1293–1302.

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang (2020). "Crisscrossed Captions: Extended Intramodal and Intermodal Semantic Similarity Judgments for MS-COCO." In: *EACL*, pp. 2855–2870.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning (2014). "Glove: Global Vectors for Word Representation." In: *EMNLP*, pp. 1532–1543.

Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan (2013). "Recognizing Text with Perspective Distortion in Natural Scenes." In: *ICCV*, pp. 569–576.

John C. Platt and Alan H. Barr (1987). "Constrained Differential Optimization." In: *NIPS*, pp. 612–621.

Golan Pundak, Tara N. Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao (2018). "Deep Context: End-to-end Contextual Speech Recognition." In: *Spoken Language Technology Workshop*, pp. 418–425.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang (2021). "RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering." In: *NAACL-HLT*, pp. 5835–5847.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). "Learning Transferable Visual Models From Natural Language Supervision." In: *ICML*, pp. 8748–8763.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). *Language Models are Unsupervised Multitask Learners*. `https://openai.com/blog/better-language-models`.

Jun Rao, Fei Wang, Liang Ding, Shuhan Qi, Yibing Zhan, Weifeng Liu, and Dacheng Tao (2022). "Where Does the Performance Improvement Come From? - A Reproducibility Concern about Image-Text Retrieval." In: *SIGIR*, pp. 2727–2737.

Nils Reimers and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: *EMNLP-IJCNLP*, pp. 3980–3990.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In: *NeurIPS*. Vol. 28, pp. 91–99.

Danilo J. Rezende and Fabio Viola (2018). "Generalized ELBO with Constrained Optimization, GECO." In: *NeurIPS Workshop on Bayesian Deep Learning*.

Anhar Risnumawan, Palaiahnakote Shivakumara, Chee Seng Chan, and Chew Lim Tan (2014). "A robust arbitrary text detection system for natural scene images." In: *Expert Systems with Applications* 41, pp. 8027–8048.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra (2021). "Can Contrastive Learning Avoid Shortcut Solutions?" In: *NeurIPS*, pp. 4974–4986.

Ties van Rozendaal, Guillaume Sautière, and Taco S. Cohen (2020). "Lossy Compression with Distortion Constrained Optimization." In: *CVPR Workshops*.

Tara N. Sainath, Rohit Prabhavalkar, Shankar Kumar, Seungji Lee, Anjuli Kannan, David Rybach, Vlad Schogol, Patrick Nguyen, Bo Li, Yonghui Wu, Zhifeng Chen, and Chung-Cheng Chiu (2018). "No Need for a Lexicon? Evaluating the Value of the Pronunciation Lexica in End-to-End Models." In: *ICASSP*, pp. 5859–5863.

Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P. Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann (2022). "Contextual Adapters for Personalized Speech Recognition in Neural Transducers." In: *ICASSP*, pp. 8537–8541.

Florian Schroff, Dmitry Kalenichenko, and James Philbin (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering." In: *CVPR*, pp. 815–823.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun (2022). "Which Shortcut Cues Will DNNs Choose? A Study from the Parameter-Space Perspective." In: *ICLR*.

Fenfen Sheng, Zhineng Chen, and Bo Xu (2018). "NRTR: A No-Recurrence Sequence-to-Sequence Model for Scene Text Recognition." In: *ICDAR*, pp. 781–786.

Baoguang Shi, Xiang Bai, and Cong Yao (2017). "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and its Application to Scene Text Recognition." In: *Transactions on Pattern Analysis and Machine Intelligence* 39, pp. 2298–2304.

Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai (2016). "Robust Scene Text Recognition with Automatic Rectification." In: *CVPR*, pp. 4168–4176.

Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai (2018). "ASTER: An Attentional Scene Text Recognizer with Flexible Rectification." In: *Transactions on Pattern Snalysis and Machine Intelligence* 41, pp. 2035–2048.

Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer (2021). "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition." In: *ICASSP*, pp. 6783–6787.

Palaiahnakote Shivakumara, Souvik Bhowmick, Bolan Su, Chew Lim Tan, and Umapada Pal (2011). "A New Gradient Based Character Segmentation Method for Video Text Recognition." In: *ICDAR*, pp. 126–130.

Ravid Shwartz-Ziv and Yann LeCun (2023). "To Compress or Not to Compress–Self-Supervised Learning and Information Theory: A Review." In: *arXiv preprint arXiv:2304.09355*.

Karen Simonyan and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *ICLR*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu (2020). "MPNet: Masked and Permuted Pre-training for Language Understanding." In: *NeurIPS*, pp. 16857–16867.

Karthik Sridharan and Sham M. Kakade (2008). "An Information Theoretic Framework for Multi-view Learning." In: *COLT*, pp. 403–414.

Bolan Su and Shijian Lu (2014). "Accurate Scene Text Recognition based on Recurrent Neural Network." In: *ACCV*, pp. 35–48.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai (2020). "VL-BERT: Pre-training of Generic Visual-Linguistic Representations." In: *ICLR*.

Guangzhi Sun, Chao Zhang, and Philip C. Woodland (2021). "Tree-Constrained Pointer Generator for End-to-End Contextual Speech Recognition." In: *Automatic Speech Recognition and Understanding Workshop*, pp. 780–787.

lya Sutskever, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks." In: *NeuRIPS*, pp. 3104–3112.

Pawel Swietojanski, Stefan Braun, Dogan Can, Thiago Fraga da Silva, Arnab Ghoshal, Takaaki Hori, Roger Hsiao, Henry Mason, Erik McDermott, Honza Silovsky, Ruchir Travadi, and Xiaodan Zhuang (2023). "Variable Attention Masking for Configurable Transformer Transducer Speech Recognition." In: *ICASSP*, pp. 1–5.

Hao Tan and Mohit Bansal (2019). "LXMERT: Learning Cross-Modality Encoder Representations from Transformers." In: *EMNLP-IJCNLP*, pp. 5099–5110.

Yonglong Tian, Dilip Krishnan, and Phillip Isola (2020a). "Contrastive Multiview Coding." In: *ECCV*, pp. 776–794.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola (2020b). "What Makes for Good Views for Contrastive Learning?" In: *NeurIPS*, pp. 6827–6839.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie (2024). "Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs." In: *arXiv preprint arXiv:2401.06209*.

Anshuman Tripathi, Jaeyoung Kim, Qian Zhang, Han Lu, and Hasim Sak (2020). "Transformer Transducer: One Model Unifying Streaming and Non-streaming Speech Recognition." In: *arXiv preprint arXiv:2010.03192*.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency (2021). "Self-supervised Learning from a Multi-view Perspective." In: *ICLR*.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic (2020). "On Mutual Information Maximization for Representation Learning." In: *ICLR*.

Michael Tschannen, Manoj Kumar, Andreas Peter Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer (2023). "Image Captioners Are Scalable Vision Learners Too." In: *NeurIPS*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals (2018). "Representation Learning with Contrastive Predictive Coding." In: *arXiv preprint arXiv:1807.03748*.

Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter (2008). *Handbook of Knowledge Representation*. Vol. 3. Elsevier. ISBN: 978-0-444-52211-5.

Ali Varamesh, Ali Diba, Tinne Tuytelaars, and Luc Van Gool (2020). "Self-Supervised Ranking for Representation Learning." In: *arXiv preprint arXiv:2010.07258*.

Ehsan Variani, Tom Bagby, Erik McDermott, and Michiel Bacchiani (2017). "End-to-End Training of Acoustic Models for Large Vocabulary Continuous Speech Recognition with TensorFlow." In: *Interspeech*, pp. 1641–1645.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All You Need." In: *NeuRIPS*, pp. 5998–6008.

Andreas Veit, Tomas Matera, Lukás Neumann, Jiri Matas, and Serge J. Belongie (2016). "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images." In: *arXiv preprint arXiv:1601.07140*.

Gaurav Verma, Vishwa Vinay, Sahil Bansal, Shashank Oberoi, Makkunda Sharma, and Prakhar Gupta (2020). "Using Image Captions and Multitask Learning for Recommending Query Reformulations." In: *ECIR*, pp. 681–696.

Haoqing Wang, Xun Guo, Zhi-Hong Deng, and Yan Lu (2022a). "Rethinking Minimal Sufficient Representation in Contrastive Learning." In: *CVPR*, pp. 16020–16029.

Kai Wang, Boris Babenko, and Serge J. Belongie (2011). "End-to-End Scene Text Recognition." In: *ICCV*, pp. 1457–1464.

Kai Wang and Serge J. Belongie (2010). "Word Spotting in the Wild." In: *ECCV*, pp. 591–604.

Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen (2020). "Cross-modal Scene Graph Matching for Relationship-aware Image-Text Retrieval." In: *WACV*, pp. 1497–1506.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei (2023). "Image as a Foreign Language: BEIT Pretraining for Vision and Vision-Language Tasks." In: *CVPR*, pp. 19175–19186.

Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork (2018). "The LambdaLoss Framework for Ranking Metric Optimization." In: *CIKM*, pp. 1313–1322.

Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao (2019). "CAMP: Cross-Modal Adaptive Message Passing for Text-Image Retrieval." In: *ICCV*, pp. 5763–5772.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao (2022b). "SimVLM: Simple Visual Language Model Pretraining with Weak Supervision." In: *ICLR*.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Ali Taylan Cemgil (2022). "A Fine-Grained Analysis on Distribution Shift." In: *ICLR*.

Ian Williams, Anjuli Kannan, Petar S. Aleksic, David Rybach, and Tara N. Sainath (2018). "Contextual Speech Recognition in End-to-end Neural Network Systems Using Beam Search." In: *Interspeech*, pp. 2227–2231.

Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell (2021). "What Should Not Be Contrastive in Contrastive Learning." In: *ICLR*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk (2021). "Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval." In: *ICLR*.

Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai (2019). "Symmetry-Constrained Rectification Network for Scene Text Recognition." In: *ICCV*, pp. 9146–9155.

Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C. Lee Giles (2017). "Learning to Read Irregular Text with Attention Mechanisms." In: *IJCAI*, pp. 3280–3286.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu (2022). "FILIP: Fine-grained Interactive Language-Image Pre-Training." In: *ICLR*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier (2014). "From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions." In: *Transactions of the Association for Computational Linguistics* 2, pp. 67–78.

Jiahui Yu, Chung-Cheng Chiu, Bo Li, Shuo-Yiin Chang, Tara N. Sainath, Yanzhang He, Arun Narayanan, Wei Han, Anmol Gulati, Yonghui Wu, and Ruoming Pang (2021a). "FastEmit: Low-Latency Streaming ASR with Sequence-Level Emission Regularization." In: *ICASSP*, pp. 6004–6008.

Tan Yu, Yi Yang, Yi Li, Lin Liu, Hongliang Fei, and Ping Li (2021b). "Heterogeneous Attention Network for Effective and Efficient Cross-modal Retrieval." In: *SIGIR*, pp. 1146–1156.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang (2021). "Florence: A new foundation model for computer vision." In: *arXiv preprint arXiv:2111.11432*.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou (2023). "When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?" In: *ICLR*.

Matthew D. Zeiler (2012). "ADADELTA: An Adaptive Learning Rate Method." In: *arXiv preprint arXiv:1212.5701*.

Donghuo Zeng, Yi Yu, and Keizo Oyama (2020). "Deep Triplet Neural Networks with Cluster-CCA for Audio-Visual Cross-modal Retrieval." In: *Transactions on Multimedia Computing, Communications, and Applications* 16.3, pp. 1–23.

Yan Zeng, Xinsong Zhang, and Hang Li (2022). "Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts." In: *ICML*, pp. 25994–26009.

Fangneng Zhan and Shijian Lu (2019). "ESIR: End-To-End Scene Text Recognition via Iterative Image Rectification." In: *CVPR*, pp. 2059–2068.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma (2021). "Optimizing Dense Retrieval Model Training with Hard Negatives." In: *SIGIR*, pp. 1503–1512.

Kun Zhang, Zhendong Mao, Quan Wang, and Yongdong Zhang (2022). "Negative-Aware Attention Framework for Image-Text Matching." In: *CVPR*, pp. 15640–15649.

Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar (2020). "Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss." In: *ICASSP*, pp. 7829–7833.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang (2018). "Asynchronous Bidirectional Decoding for Neural Machine Translation." In: *AAAI*, pp. 5698–5705.

Ying Zhang and Huchuan Lu (2018). "Deep Cross-Modal Projection Learning for Image-Text Matching." In: *ECCV*, pp. 686–701.

Ding Zhao, Tara N. Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang (2019). "Shallow-Fusion End-to-End Contextual Biasing." In: *Interspeech*, pp. 1418–1422.

Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun (2017). "Multi-view Learning Overview: Recent Progress and New Challenges." In: *Inf. Fusion* 38, pp. 43–54.

Long Zhou, Jiajun Zhang, and Chengqing Zong (2019). "Synchronous Bidirectional Neural Machine Translation." In: *Transactions of the Association for Computational Linguistics* 7, pp. 91–105.

Yingying Zhu, Cong Yao, and Xiang Bai (2016). "Scene Text Detection and Recognition: Recent Advances and Tuture Trends." In: *Frontiers of Computer Science* 10, pp. 19–36.

Yongshuo Zong, Oisin Mac Aodha, and Timothy M. Hospedales (2023). "Self-Supervised Multimodal Learning: A Survey." In: *arXiv preprint arXiv:2304.01008*.

Summary

In this thesis, we work on multi-modal learning problems and algorithms. To that end, we center our investigations around three modalities: (i) *audio*, (ii) *image(s)*, and (iii) *text*. We focus on three evaluation tasks to study these modalities: (i) *automatic speech recognition* (ASR), (ii) *scene text recognition* (STR), and (iii) *image-caption retrieval* (ICR). We provide novel methods and insights into two directions: multi-modal sequence modeling and multi-modal representation learning.

The first part of the thesis consists of two chapters focussing on multi-modal sequence modeling. In Chapter 2, we propose an efficient online hard negative mining approach for contextual speech recognition: *approximate nearest neighbour phrase* (ANN-P) *mining*. The goal of ANN-P mining is to improve the model's ability to disambiguate between similar-sounding phrases and hence the prediction performance of the ASR model. We show that by mining hard negative phrases from the latent space of the context encoder, up to 7% relative word error rate reduction can be achieved for the personalized test data in streaming scenarios.

In Chapter 3, we introduce the *bidirectional scene text transformer* (Bi-STET). Bi-STET is a bidirectional STR method. However, in contrast to other bidirectional STR methods, Bi-STET uses a single decoder for both decoding directions. Due to the non-recurrent inductive bias of the transformer, it becomes possible to condition the decoding direction of the output sequence at the input level. We show that Bi-STET outperforms methods using two decoders and performs on par or outperforms other state-of-the-art STR methods.

The second part of the thesis consists of three chapters focussing on contrastive multi-modal representation learning, specifically for images and text. In Chapter 4, we explore the generalization of metric learning functions to the ICR task. We find that the lessons from metric learning do not generalize to the ICR task. To understand these empirical findings, we introduce the *counting contributing samples* (COCOS) method. By using the COCOS method we show that the best performing metric learning loss takes only one hard negative into account when computing the gradient. Additionally, the COCOS method suggests that underperforming losses take too many (uninformative) negative samples into account when computing the gradient.

In Chapter 5, we investigate the problem of predictive feature suppression for resource-constrained ICR methods. To reduce predictive feature suppression, we introduce *latent target decoding* (LTD). LTD is a non-recurrent reconstruction objective that reconstructs the input caption in the latent space of a sentence encoder. We show that LTD reduces predictive feature suppression, by outperforming ICR methods that are solely optimized with a contrastive loss. Furthermore, we find that implementing LTD as an optimization constraint is more effective than as a dual loss. Finally, we show that LTD can be combined with different ICR methods and contrastive losses.

In Chapter 6, we investigate a different problem in image-text representation learning: *shortcut learning*. For image-text representation learning with contrastive InfoNCE-based optimization objectives, it remains unclear if those losses are suitable to capture all task-relevant information or if they rely on a shortcut. To that end, we introduce the *synthetic shortcuts for vision-language* (SVL) framework: a training and evaluation framework that allows us to inject of synthetic shortcuts into image-text data. We show that, by using the SVL framework, contrastive image-text methods predominantly learn shortcut features when a shortcut is present in the training data. Hence, the InfoNCE loss is not sufficient to learn to represent all task-relevant information in the data. As a next step, we examine two shortcut reduction methods on the SVL framework. We find that both methods partially mitigate shortcut learning when training and evaluating with our SVL framework.

Samenvatting

In dit proefschrift onderzoeken we multi-modale leerproblemen en algoritmen. Daartoe richten we ons onderzoek op drie modaliteiten: (i) *audio*, (ii) *afbeeldingen* en (iii) *tekst*. We concentreren ons op drie evaluatietaken om deze modaliteiten te bestuderen: (i) *automatic speech recognition* (ASR), (ii) *scene text recognition* (STR) en (iii) *image-caption retrieval* (ICR). We bieden nieuwe methoden en inzichten in twee richtingen: multi-modaal sequentiemodellering en het leren van multi-modale representaties.

Het eerste deel van het proefschrift bestaat uit twee hoofdstukken die zich richten op multimodaal sequentiemodellering. In Hoofdstuk 2 introduceren we een efficiënt, online *hard negative mining* methode voor contextuele ASR: *approximate nearest neighbour phrase* (ANN-P) *mining*. Het doel van ANN-P mining is het verbeteren van het vermogen van het ASR model om onderscheid te maken tussen gelijk klinkende zinsdelen/woorden, en dus de transcripties van het model te verbeteren. We laten zien dat, door de *hard negatives* te verkrijgen uit de *latent space* van de *context encoder*, tot 7% relatieve reductie van de *word error rate* kan worden bereikt voor het gepersonaliseerde deel van de testdata in streamingscenario's.

In Hoofdstuk 3 introduceren we *bidirectional scene text transformer* (Bi-STET). Bi-STET is een bidirectionele STR-methode. Echter, in tegenstelling tot andere bidirectionele STR-methoden maakt Bi-STET gebruik van één decoder voor beide decodeerrichtingen. Vanwege de non-recurrente *inductive bias* van de *transformer*, wordt het mogelijk om de decodeerrichting van de output op inputniveau te conditioneren. We laten zien dat Bi-STET beter presteert dan methoden die twee decoders gebruiken. Daarnaast tonen we aan dat Bi-STET op hetzelfde niveau of beter presteert dan andere *state-of-the-art* STR-methoden.

Het tweede deel van het proefschrift bestaat uit drie hoofdstukken die zich richten op het leren van contrastieve multimodale representatie, specifiek voor afbeeldingen en tekst. In Hoofdstuk 4 onderzoeken we de generalisatie van *metric learning* functies naar de ICR-taak. We constateren dat de lessen uit *metric learning* niet generaliseren naar de ICR-taak. Om deze empirische bevindingen te begrijpen, introduceren we de *counting contributing samples* (COCOS) methode. Door de COCOS-methode te gebruiken laten we zien dat de best presterende optimalisatie functie met slechts één hard-

negatieve rekening houdt bij het berekenen van de gradiënt. Bovendien suggereert de COCOS-methode dat ondermaats presterende optimalisatie functies te veel (niet-informatieve) negatieve samples meenemen bij het berekenen van de gradiënt.

In Hoofdstuk 5 onderzoeken we het probleem van de onderdrukking van voorspellende *features* in de input data voor ICR-methoden met beperkte trainingsmiddelen. Om de onderdrukking van voorspellende *features* te verminderen, introduceren we *latent target decoding* (LTD). LTD is een niet-recurrente reconstructiedoel dat de input tekst in de *latent space* van een zinsencoder reconstrueert. We laten zien dat LTD de onderdrukking van voorspellende *features* vermindert, door beter te presteren dan ICR-methoden die uitsluitend zijn geoptimaliseerd met een contrastieve optimalisatie functie. Bovendien vinden we dat het implementeren van LTD als een optimalisatie voorwaarde effectiever is als een *dual* optimalisatie functie. Ten slotte laten we zien dat LTD kan worden gecombineerd met verschillende ICR-methoden en contrastieve optimalisatie functies.

In Hoofdstuk 6 onderzoeken we een ander probleem bij het leren van afbeelding-tekst representaties: het *shortcut learning* probleem. Voor het leren van contrastieve afbeelding-tekst methoden met de op InfoNCE-gebaseerde optimalisatie functie, blijft het onduidelijk of deze optimalisatie functies geschikt zijn om alle taakrelevante informatie vast te leggen, of dat ze afhankelijk worden van een *shortcut*. Daartoe introduceren we *synthetic shortcuts for vision-language* (SVL): een trainings- en evaluatie-kader dat ons in staat stelt synthetische *shortcuts* in afbeelding-tekst data te injecteren. We laten zien dat, door gebruik te maken van het SVL framework, contrastieve afbeelding-tekst methoden voornamelijk *shorcuts* leren wanneer die aanwezig zijn in de training data. We concluderen dat de InfoNCE optimalisatie functie niet voldoende is om alle taakrelevante informatie in de gegevens weer te geven. Als volgende stap onderzoeken we twee methoden voor het verminderen van *shortcuts* binnen het SVL-kader. We constateren dat beide methoden het leren van *shortcuts* gedeeltelijk verminderen tijdens het trainen en evalueren met het SVL-kader.