On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition

by

Yonatan Belinkov

B.Sc., Mathematics, Tel Aviv University (2009)
M.A., Arabic and Islamic Studies, Tel Aviv University (2014)
S.M., Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
May 21, 2018

Certified by . . . .
James R. Glass
Senior Research Scientist
Computer Science and Artificial Intelligence Laboratory
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Theses

On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition

by

Yonatan Belinkov

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2018, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Language technology has become pervasive in everyday life. Neural networks are a key component in this technology thanks to their ability to model large amounts of data. Contrary to traditional systems, models based on deep neural networks (a.k.a. deep learning) can be trained in an end-to-end fashion on input-output pairs, such as a sentence in one language and its translation in another language, or a speech utterance and its transcription. The end-to-end training paradigm simplifies the engineering process while giving the model flexibility to optimize for the desired task. This, however, often comes at the expense of model interpretability: understanding the role of different parts of the deep neural network is difficult, and such models are sometimes perceived as "black-box", hindering research efforts and limiting their utility to society.

This thesis investigates what kind of linguistic information is represented in deep learning models for written and spoken language. In order to study this question, I develop a unified methodology for evaluating internal representations in neural networks, consisting of three steps: training a model on a complex end-to-end task; generating feature representations from different parts of the trained model; and training classifiers on simple supervised learning tasks using the representations. I demonstrate the approach on two core tasks in human language technology: machine translation and speech recognition. I perform a battery of experiments comparing different layers, modules, and architectures in end-to-end models that are trained on these tasks, and evaluate their quality at different linguistic levels.

First, I study how neural machine translation models learn morphological information. Second, I compare lexical semantic and part-of-speech information in neural machine translation. Third, I investigate where syntactic and semantic structures are captured in these models. Finally, I explore how end-to-end automatic speech recognition models encode phonetic information. The analyses illuminate the inner workings of end-to-end machine translation and speech recognition systems, explain how they capture different language properties, and suggest potential directions for improving them. I also point to open questions concerning the representation of other linguistic properties, the investigation of different models, and the use of other analysis methods. Taken together, this thesis provides a comprehensive analysis of internal language representations in deep learning models.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist
Computer Science and Artificial Intelligence Laboratory

להוריי, באהבה ובהערכה

Acknowledgments

To my advisor, Jim Glass, thank you for welcoming me to SLS. I'm especially lucky to have found an advisor who gives me freedom to pursue my ideas, and numerous side projects. At the same time, thank you for pushing me back on track from time to time and making sure this PhD comes to an end. I value your broad perspective and long-term outlook, but most of all, your kindness and consideration have made this a pleasant journey.

I would like to thank my committee members, Tommi Jaakkola and Peter Szolovits, for providing helpful feedback and advice. Thank you for bearing with my tardiness. Your comments have made this thesis more complete, and gave me plenty of ideas for future research.

I feel extremely fortunate to work in the Spoken Language Systems group, a welcoming environment where I could talk to people on anything research and life, hang out over ice cream or beer, and simply be happy to come to the office every day. Thank you to past and present members of SLS for making this such a great environment: Ann, Chen, Daniel, Dave, Di-Chia, Ekapol, Hao, Hongyin, Jen, Mandy, Maryam, Michael, Mitra, Najim, Patrick, Ramy, Sameer, Scott, Sree, Stephanie, Stephen, Suwon, Tianxing, Tuka, Victor, Wei-Ning, Xue, Yu, and Yu-An, as well as all the MEng and undergraduate students I interacted with over the years. Much appreciation and gratitude to Marcia Davidson, our unique administrative assistant, for all her assistance and for making our life much easier.

I gratefully acknowledge the support of the Qatar Computing Research Institute (QCRI) during the course of my PhD. In addition to financial support, I would like to thank my many collaborators at QCRI: Ahmed Abdelali, Ahmed Ali, Alberto, Alessandro, Fahim, Giovanni, Hamdy, Hassan, Kareem, Lluís, Mohamed Eldesouki, Nadir, Salvatore, Sameer, and Stephan. Special thanks to Alessandro Moschitti for leading our research collabora-

tion and for his kind advice. I'm especially grateful to the MT team: Fahim Dalvi, Nadir Durrani, and Hassan Sajjad, with whom I collaborated closely on large parts of this thesis. Thank you for teaching me so much about MT; I look forward to more days and nights of chasing deadlines.

Many thanks to other collaborators I was lucky to work with during my PhD: Adam, Alex, Avi, Benjamin, Kfir, Maxim, Moshe, Roee, Yoav, and Yonatan Bisk. Thank you also to my IBM collaborators for a wonderful summer: Einat, Ofer, Yossi, and others at IBM Research.

Moving to Cambridge and leaving my previous life in Israel was not a simple step. Fortunately, I found an amazing community of Israelis in the Boston area, friends who became family for every occasion, good or bad. Some have left and others came, but all of you are part of what makes this place home.

To Niva, thank you for standing with me every step of the way. It's been a long, and sometimes hard ride, and there were times I didn't think I'll make it to the finish line. Your constant love and support have made this possible. I couldn't have done this without you.

To my parents, thank you for supporting me in so many ways. The foundations you instilled in me since a young age have directly impacted my success. I dedicate this thesis to you, in love and appreciation.

Bibliographic Note

Parts of this thesis are based on prior peer-reviewed publications. Chapter 2 is mainly based on [32], with additional experiments from [83]. I added more experiments and analysis in this chapter. Chapter 3 is based on [33], with additional experiments and more details on the experimental design. The work presented in Chapter 5 was published in [29]. Some additional experiments were included in this chapter.

Most of the code developed in this thesis is available at the following repositories:

`https://github.com/boknilev/nmt-repr-analysis`

`https://github.com/boknilev/asr-repr-analysis`

Contents

List of Figures

List of Tables

List of Abbreviations

Notation	Description	Page List
AI	Artificial Intelligence	35, 36, 39
CNN	convolutional neural network	16, 51, 52, 56, 61, 69, 70, 79, 154
CTC	connectionist temporal classification	33, 61, 62, 132, 133, 135, 143, 149
DTW	dynamic time warping	59
EM	expectation maximization	55
FFT	fast Fourier transform	59
GMM	Gaussian mixture model	60
GRU	gated recurrent unit	57
HMM	hidden Markov model	59, 61

Notation	Description	Page List
LDA	latent Dirichlet allocation	51
LPC	linear predictive coding	59
LRP	layer-wise relevance propagation	46
LSA	latent semantic analysis	49, 51
LSTM	long short-term memory	45, 47, 51, 57, 61, 69–71, 103, 136
MFCC	Mel-frequency cepstral coefficient	45, 53, 61, 154
NLP	natural language processing	44, 46, 48, 52, 103, 115, 167
OOV	out-of-vocabulary	52, 74, 109
PBMT	phrase-based machine translation	114
PMB	Groningen Parallel Meaning Bank	99, 111
POS	part-of-speech	15–17, 25–27, 39, 43, 52, 63, 64, 67, 69, 70, 72–83, 85, 90, 94–106, 111, 117, 121, 151, 157, 160, 176
RecNN	recursive neural network	51
ReLU	rectified linear unit	70, 136, 137

Notation	Description	Page List
RNN	recurrent neural network	37, 45, 46, 51, 56–58, 61, 62, 87, 98, 103, 120, 123, 132, 134, 136, 141
SEM	semantic	17, 18, 25–27, 94–107, 109–111, 151, 159, 160
seq2seq	sequence-to-sequence	33, 37, 47, 56, 58, 61, 62, 64, 132–134, 149
SGD	stochastic gradient descent	70
SVD	singular value decomposition	49
WER	word error rate	137

Glossary

Notation	Description	Page List
Adam	An adaptive optimization method	71
attention	Mechanism for aligning source and target inputs in sequence-to-sequence (seq2seq) models	10, 16, 37, 46, 51, 52, 58, 61, 62, 68, 70, 83–86, 88, 90, 98, 120, 133, 134
cepstral analysis	A common procedure for deconvolution of a speech signal	53, 59
DeepSpeech2	End-to-end ASR model based on connectionist temporal classification (CTC)	20–22, 26, 27, 135–144, 147, 179, 180
dropout	Method for regularizing neural networks	70, 137
F_1	An evaluation metric defined as the harmonic mean of precision and recall	17, 21, 107, 108, 145–147
feed-forward		137

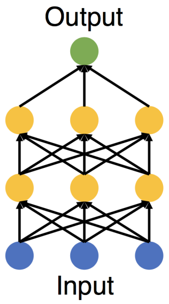
Notation	Description	Page List
formant	A natural frequency or resonance of a speech signal	59
Hamming	A window function often used in speech processing	135
highway	Type of connection in neural networks	70
lemma	A dictionary item	65
morpheme	A meaningful morphological unit	50, 65, 66
phone	A speech sound	21, 26, 132–134, 136–139, 143–150
phoneme	An abstract speech class that carries meaning in a specific language	46, 50, 59, 60, 132–134
Softmax	Normalization layer used in classification tasks	57, 63, 70, 137

Chapter 1

Introduction

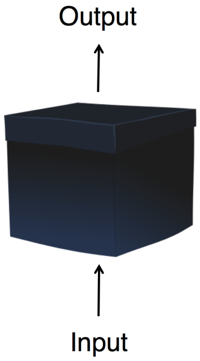
"Whatever you cannot understand,
you cannot possess."
— Johann Wolfgang von Goethe

Language technology has become pervasive in everyday life, powering applications like Apple's Siri, Google's Assistant or Amazon's Alexa. These systems have grown in popularity in recent years thanks to advances in Artificial Intelligence (AI) technology. The new wave of AI stands on three pillars: massive amounts of data, high-performance computing resources, and computational models and algorithms that have the capacity to utilize these resources. Artificial neural networks are a key ingredient in the success of AI systems in general, and language technology in particular. These computational models are excellent "learners"—they can be trained on large amounts of examples generated by humans and learn the pertinent information in the data they are trained on. This machine learning paradigm enables AI systems to answer questions, recognize human speech, and translate sentences between multiple languages.



A “deep” neural network

An important property of artificial neural networks is the ability to train them in an end-to-end fashion, i.e., the entire system is based on one model that is optimized to solve a task of interest (e.g., translate sentences). Whereas traditional systems contain multiple modules that are built separately and only combined at a later stage, end-to-end systems are trained jointly on the final task by stacking multiple layers of artificial neural networks in one model, also known as “deep learning”. The main advantages of deep learning models are their simplicity and the fact that the entire model is optimized for the end task. However, such models are much more difficult to interpret than their predecessors. It is not clear what the role of different components is, how they interact, and what kind of information they learn during the training process. Consequently, systems based on neural networks are often thought of as a “black-box”—they map inputs to outputs, but the internal machinery is opaque and difficult to interpret.



A neural network?

The lack of interpretability has major implications for the adoption and further development of AI systems. First, gaining a better understanding of these systems is necessary for improving their design and performance. In current practice, their development is often limited to a trial-and-error process whereby engineers tweak a part of the system, retrain it on a large dataset, and measure the final performance, without gaining a real understanding of what the system has learned. More importantly, as more and more AI systems are being integrated in our daily lives, we need to make sure we can understand and explain their automatic predictions. Interpretability is important for guaranteeing fairness and accountability in AI systems—if we do not understand the systems we cannot expect them to be fair to all members of our society. Nor can we expect the public to be confident in relying on such systems.

Much work in deep learning for language is concerned with the performance on some end task. A common scenario is to propose new neural network architectures and compare their performance on a benchmark dataset. For example, different architectures for neural machine translation have been proposed and evaluated on standard machine translation datasets.[1] The common research process may be described as an iterative process. First, researchers design a new neural network architecture. Then, they train the system and evaluate its performance on some task. If the performance is not satisfactory, the researchers change the architecture and re-train and evaluate the system. This process is repeated until sufficiently good performance is achieved. Figure 1-1 illustrates this process.

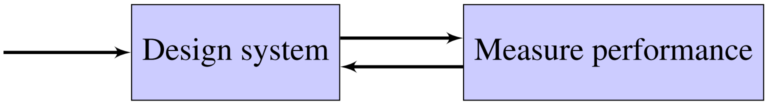


Figure 1-1: Common practice in deep learning research iterates between designing an end-to-end system and evaluating its performance on the end task.

The limitations of the above process have been recognized by the research community. In an effort to gain more confidence in the quality of different models, researchers often evaluate them on multiple downstream tasks. For instance, different methods for obtaining vector representations of words, also known as word embeddings, have been proposed, and their quality may be evaluated on a variety of tasks [22]. Another example is sentence embeddings, which are often evaluated on sentence classification and sentence similarity tasks [80, 116, 147, 180, 201].

This approach still does not provide much insight about the underlying model; to a large extent, the neural network remains a black-box. This thesis investigates what kind of linguistic information is captured in such models. It is focused on two key problems in human language technology: machine translation and speech recognition. Long recognized as fundamental problems in artificial intelligence and computational linguistics, the recent

[1] Primary examples include recurrent neural network (RNN) sequence-to-sequence models [318] and their attentional variants [16, 221], convolutional sequence-to-sequence models [122], and fully-attentional models [327], although numerous variants have been proposed in recent years.

years have witnessed great progress in research and development of systems for recognizing human speech and translating foreign language. If we are ever to achieve anything the Hitchhiker's "Babel fish", solving machine translation and speech recognition is a key step.

if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language
— The Hitchhiker's Guide to the Galaxy

In this introduction, I first comment on terminological issues regarding analysis and interpretation in machine learning (Section 1.1). Then I present the high-level methodological approach used throughout this thesis for analyzing deep learning models for language (Section 1.2). Section 1.3 surveys related work on the analysis of neural networks in language and speech processing. This section aims to provide a brief summary of other analysis methods that have been considered in the literature. Much of this thesis is concerned with representations of language as they are learned by end-to-end deep learning models. To properly situate this within the broader work on representing language, I provide in Section 1.4 a short overview of language representations as they are used in human language technology, focusing on distributed, vector-based representations, sometimes referred to as "embeddings" in deep learning parlance. The following two sections provide the necessary background on machine translation (Section 1.5) and speech recognition (Section 1.6). These two tasks have a rich and intertwined history, which I briefly summarize before laying down the formal probabilistic models that are commonly used for these tasks. In both cases, I define the neural network approaches that will be studied in the remainder of this thesis. Finally, I provide a summary of contributions in Section 1.7.

Thesis roadmap

Before proceeding, it may be helpful to provide a roadmap of the thesis. The work described in this thesis can be viewed from several perspectives. First, in terms of applications, I study two fundamental tasks in human language technology. The bulk of the thesis is concerned with analyzing neural machine translation (Chapters 2–4). Chapter 5 extends the same approach to automatic speech recognition as a proof-of-concept for the generalizability of the ideas. Second, in terms of the linguistic information that is being analyzed, the studies reported in this thesis target the representation of different linguistic units. Chapters 2 and 3 deal with properties of individual words, while Chapter 4 studies relations between pairs of words, a basic notion of structure. Chapter 5 goes down to the phonetic level and studies speech representations. Third, these language representations are investigated through specific core language and speech processing tasks: part-of-speech (POS) and morphological tagging (Chapter 2), semantic tagging (Chapter 3), syntactic and semantic dependency labeling (Chapter 4), and phone classification (Chapter 5). Taken together, this thesis provides a multi-faceted analysis of internal representations in deep learning models for language and speech processing.

1.1 Interpretability, Explainability, Transparency, and What This Thesis Is Not About

Interpretability, explainability, transparency, explainable AI (XAI) — these and other terms have been used, somewhat interchangeably, in the context of work on deep learning, and more broadly machine learning and AI. At present there seems to be no consensus on their precise definition and application to the study of AI systems. A short consideration of aspects of terminology will help situate this thesis in the broader work on interpretability in AI.[2]

Miller [239] surveys a range of work on explanation in the social sciences with relevance to AI. He takes a rather narrow view of interpretability, defining it as "the degree to which an observer can understand the cause of a decision". To him, interpretability is the same as explainability, and the two are different from explicitly explaining decisions for given examples. While explaining specific model predictions is obviously important in

[2] See [94, 212], as well as the online book by Christoph Molnar for more reflections and references: `https://christophm.github.io/interpretable-ml-book/`.

work on deep learning for language, this is not the goal of this thesis. However, relevant work along these lines is briefly mentioned in Section 1.3.2.

Doshi-Velez and Kim [94] define interpretability more generally as "the ability to explain or to present in understandable terms to a human". Notice that their definition does not refer to *decisions*. Lipton [212] recognizes that "interpretability is not a monolithic concept, but in fact reflects several distinct ideas". He contrasts *transparency*, which is concerned with how the model works, with *post-hoc explanations*. Transparency, however, may mean different things to different stakeholders: developers, users, or the society as a whole [337]. The methods and experiments in this thesis will be primarily of interest to machine learning researchers and practitioners, especially those focusing on language and speech processing. Researchers from closely related disciplines, namely linguists and cognitive scientists, may also be interested in the methodology and some of the results on what kind of linguistic information is learned by artificial neural networks.

Another important criterion is the level of analysis. Interpretability and transparency can operate at a *local* level, providing explanations for a particular decision [94, 337], or at a *global* level, forming a general understanding of the model or system. Such work may also be further categorized as applied to the entire model, certain model parts, or the underlying algorithms [212]. This thesis aims to provide a better understanding of the different parts and modules in deep learning models for language (striving for decomposibility, in the sense of [212]). In terms of the levels of analysis put forth by Marr and Poggio [226], it is concerned mainly with the algorithmic level: what mechanisms and representations are used by deep learning models of language.

Finally, there has been some debate in the community regarding the need for interpretability.[3] Arguments in favor include goals like accountability, trust, fairness, safety, and reliability. Arguments against typically stress performance as the most important

[3] For example, a NIPS 2017 debate: `https://www.youtube.com/watch?v=2hW05ZfsUUo`.

desideratum. Without dwelling on these debates, I outline in the following section my high-level approach for analyzing deep learning models, arguing that it sets a better and more informed research process. The reader can decide if this thesis meets this goal.

1.2 Methodological Approach

The methodology advocated in this thesis aims to depart from the common deep learning research process, which typically iterates between designing an end-to-end system and evaluating its performance on the end-task (Figure 1-1). The key idea is to utilize supervised learning tasks to probe the internal representations in end-to-end models. The first step is to train an existing end-to-end system such as a neural machine translation system. Then, the trained system is used for generating feature representations. Finally, a separate classifier is trained, and evaluated, on predicting some linguistic property using the generated representations. The classifier's performance reflects the quality of the representations for the given task, and by proxy, it also reflects the quality of the original model. This process is illustrated in Figure 1-2.

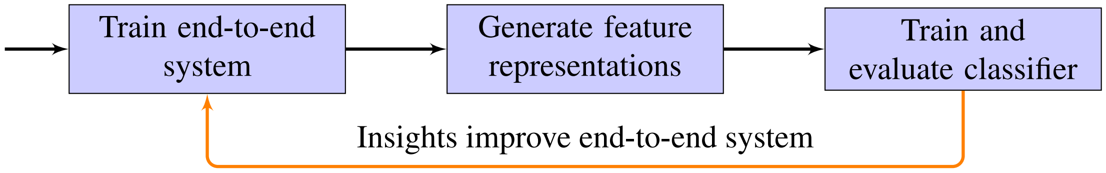

Train end-to-end system
Generate feature representations
Train and evaluate classifier
Insights improve end-to-end system

Figure 1-2: Proposed methodology for alternating between training neural models and evaluating the learned representations on specific properties. First, an end-to-end system is trained. Second, feature representations are generated with the trained model. Third, the quality of the representations is evaluated by training a classifier on a supervised learning task. Finally, insights from the analysis can be used to improve the original end-to-end system.

Formally, let $f(\cdot;\phi)$ denote an end-to-end neural model that maps inputs x to outputs y and is parameterized by ϕ. Denote by $\bar{f}(x;\phi)$ some *internal* representation of x obtained at an intermediate step during the computation of $f(x;\phi)$. Define a separate classifier $g(\cdot;\psi)$ that takes the internal representation $\bar{f}(x;\phi)$ as input and maps it to an output label z. At the first step, $f(\cdot;\phi)$ is trained on examples $\{x, y\}$ and ϕ is updated using back-propagation [288]. At the second step, $\bar{f}(\cdot;\phi)$ generates internal feature representations. At the last step, $g(\cdot;\psi)$ is trained on examples $\{\bar{f}(x), z\}$ and ψ is updated. Crucially, at this step ϕ is not being updated in order to maintain the original representations. In other words, back-propagation is applied only to g and not to f.

This procedure can also be cast in informational theoretic terms. Let $h = \bar{f}(x;\phi)$ and consider the cross-entropy objective function over a training set $\{\tilde{h}, \tilde{z}\}$:

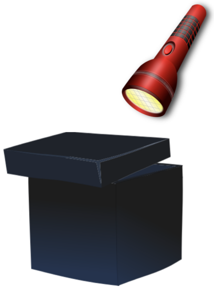
$$-\sum_{\tilde{h},\tilde{z}} \log P_{\psi}(\tilde{z}|\tilde{h}) \tag{1.1}$$

This is an unbiased estimator of the conditional entropy, so minimizing the cross-entropy is trying to minimize the conditional entropy:

$$H(\mathrm{z}|\mathrm{h}) = -\mathbb{E}_{h,z\sim P}[\log P(z|h)] \tag{1.2}$$

Now, recall the relation between mutual information and conditional entropy, $I(\mathrm{h},\mathrm{z}) = H(\mathrm{z}) - H(\mathrm{z}|\mathrm{h})$, and note that $H(\mathrm{z}) = -\mathbb{E}_{z\sim P}[\log P(z)]$ is constant (labels z are given and thus the marginal $P(z)$ is known and remains unchanged). This means that our procedure attempts to maximize the mutual information between the internal representation h and the linguistic property z.

The power of this approach stems from the possibility of comparing representations from different end-to-end models, or from different parts of a given model. For instance, one could compare representations from different layers of a deep neural network. Moreover, evaluating representation quality on different classification tasks provides a window onto what kind of linguistic information is captured in the neural network.



Opening the black box

As a concrete example, consider neural machine translation as the end-to-end model to study, f, and suppose we are interested in finding out which parts of the model store information about parts-of-speech. The neural machine translation model is trained on parallel sentences (x, y), where x is a source sentence and y is a target sentence. Then, word representations are generated by running source sentences through the encoder and obtaining the encoded vector representations at the top layer of the encoder, $\bar{f}(x; \phi)$. These representations are input to a classifier $g(\cdot; \psi)$ that predicts a POS tag for every word. The performance of the classifier is evaluated on a POS tagging test set.

There is one last important component to this approach. If the analysis is successful, then the results should be useful and applicable. Therefore, the final step is to improve the original end-to-end model based on insights from the analysis. This step aims to close the loop and connect the analysis back to the design of the original end-to-end system, as illustrated in Figure 1-2. This thesis includes one such success story in Chapter 2.

Finally, a note on potential limitations of the outlined methodology. The approach relies on the assumption that the performance of the classifier g reflects the quality of the end-to-end model f for the classification task. This is a reasonable assumption since the input to the classifier are the representations generated by the end-to-end-model, which are trained for the original task. Nevertheless, it is possible, even if unlikely, that the classifier performs well by chance and not because the representations need to learn useful information about the task. Future work may investigate evidence for a *causal* relationship between the complex end-to-end task and the simpler linguistic property. Another potential concern is that the classifier is either too weak or too strong. If it is too weak, then the representations may contain information that the classifier cannot extract, and the results might reflect too negatively on the quality of the end-to-end model. If the classifier is too strong, then it may be able to find patterns that the end-to-end model cannot utilize. The majority of the experiments in this work are conducted with a one hidden layer neural network. This setting aims to strike a balance in classifier power. In several cases, other classifiers are compared. The results typically show that stronger classifiers perform better in absolute terms, as expected. More importantly, however, experimenting with different classifiers leads to consistent relative trends when comparing different inputs, such as representations from different layers.

1.3 Related Analysis Work

The past few years have seen significant interest in analyzing neural networks in written and spoken language processing tasks. Much of the work in the field has been concerned with asking what kind of linguistic information is captured in these models. This question is at the core of the thesis and so I first discuss related work that tries to answer it more or less directly. Then I review other work that sheds light on different aspects of deep learning for language.

1.3.1 What linguistic information is captured in deep learning models

This question can be studied along three dimensions: which objects in the neural network are being investigated, what kind of linguistic information is sought, and which methods are used for conducting the analysis.

In terms of the object of study, previous work has looked for linguistic information in different neural network components. In natural language processing (NLP), this includes word embeddings [138, 188, 279], RNN hidden states or gate activations [102, 278, 306, 335, 345], and sentence embeddings [2, 3, 47, 81, 106, 117]. In speech processing, researchers have analyzed layers in deep neural networks for speech recognition [243, 251, 252], and different speaker embeddings [333]. Others have analyzed joint language-vision [123] or audio-vision models [5, 73, 142].

Different kinds of linguistic information have been analyzed, starting from basic properties like sentence/utterance length, word presence, or simple word order [2, 3, 73, 81, 117], through morphological [279, 330], syntactic [81, 101, 137, 188, 210, 278, 279, 306, 322], and semantic information [101, 102, 279]. Phonetic/phonemic information [251, 252, 335] and speaker information [251, 333] have been studied in neural network models for speech, as well as in joint audio-visual models [5].

Methodologically, many studies look for correspondences or associations between parts of the neural network and certain properties. This may be computed directly, for instance by computing the correlation between long short-term memory (LSTM) cell activations and Mel-frequency cepstral coefficient (MFCC) acoustic features [345], or indirectly, by defining discrimination tasks based on activations [5, 57]. A more common approach is to predict certain linguistic properties from activations of the neural network [2, 106, 117, 123, 188, 252, 278, 279, 306, 333].[4] In this thesis, I follow a similar approach for analyzing end-to-end models for machine translation and speech recognition.

1.3.2 Other analysis methods

Visualization Visualization has been a valuable tool for analyzing neural networks in the language domain and beyond. Early work visualized hidden unit activations in RNNs trained on an artificial language modeling task, and observed how they correspond to cer-

[4] A similar method has been used to analyze hierarchical structure in neural networks trained on arithmetic expressions [155, 328].

tain grammatical relations such as agreement [103]. Much recent work has focused on visualizing activations on specific examples in modern neural networks for language [103, 168, 173, 278] and speech [251, 345]. The attention mechanism that originated in work on neural machine translation [16] also lends itself to a natural visualization.[5]

Another line of work computes various saliency measures to attribute predictions to input features. The important or salient features can then be visualized in selected examples [12, 203, 247, 248, 317]. For instance, layer-wise relevance propagation (LRP), which propagates a measure of relevance down the network [37] has been applied to several language processing tasks [10, 11], including neural machine translation [90].[6]

An instructive visualization technique is to cluster neural network activations with respect to some linguistic properties. Early work has clustered RNN activations showing that they organize in lexical categories [101, 102]. Similar techniques have been followed by others; recent examples include clustering of sentence embeddings in an RNN encoder trained in a multi-task learning scenario [47], and phoneme clusters in a joint audio-visual RNN model [5].

Challenge sets Another approach for analyzing deep learning models is to evaluate their performance on carefully constructed examples, known as challenge sets or test suites. Following work in NLP [196] and machine translation [178], a number of such suites have been manually constructed to evaluate neural machine translation performance on a range of linguistic phenomena [23, 49, 161]. These manually-crafted datasets present high-quality examples that enable fine-grained evaluation of translation quality. However, they are usually quite small.[7] An alternative approach is to generate a large number of

[5] Sometimes the use of attention is even motivated by a desire "to incorporate more interpretability into the model" [191].

[6] Many of the visualization methods are adapted from the vision domain, where they have been extremely popular; see [356] for a survey.

[7] Typical sizes are in the hundreds [23, 161] or thousands [49].

examples programmatically to study specific phenomena, such as morphology [50], syntax [301], or word sense disambiguation [285]. Such datasets offer a less-nuanced evaluation, but they allow for large-scale experiments and a more statistically valid evaluation.[8]

The challenge set evaluations can be seen as complementary to the approach taken in this thesis, which is concerned with the quality in "the average case", as the experiments are conducted on standard test sets that are randomly sampled from the data. The limitation is that the results may not generalize to edge cases such as in carefully constructed challenge sets. The advantage is that the results are more likely to capture the performance in the typical case.

Explaining predictions Explaining specific model predictions is recognized as a desideratum for increasing the accountability of machine learning systems [95].[9] However, explaining why a deep, highly non-linear neural network makes a certain prediction is not trivial. One possibility for achieving this is to ask the model to generate explanations along with its primary prediction [353, 358]. The shortcoming of this approach is that it requires manual annotations of explanations, which can be difficult to collect. An alternative approach is to use parts of the input as explanations in a classification scenario [197], or input-output associations in a sequence-to-sequence learning scenario [7]. Another interesting recent direction, explored in the vision domain, is to identify influencing training examples for a particular prediction [187]. Other work considered learning textual-visual explanations from multimodal manual annotations [271].

[8] Similarly motivated datasets have been constructed to evaluate models in other tasks than machine translation, such as subject-verb agreement in LSTM language models or classifiers [136, 210], or compositionality in sequence-to-sequence (seq2seq) learning [192] and language inference tasks [84].

[9] See also [282] for a recent overview of explanation methods in deep learning that takes a very broad view of explanation, including saliency and other attribution methods.

Adversarial examples Understanding a model requires also an understanding of its failures. Despite their success in many tasks, machine learning systems can also be very sensitive to malicious attacks or adversarial examples [36, 128, 232, 319]. In the machine vision domain, small changes to the input image can lead to misclassification, even if such images are indistinguishable by humans [128, 319]. Adversarial examples can be generated using access to model parameters (white-box attacks) or without such access (black-box attacks) [217, 255, 267, 269].

Adversarial examples have also begun to be explored in NLP. A few white-box attacks look for important text edit operations that will fool a classifier [99, 207, 268, 294]. Others have considered black-box adversarial examples for text classification [118] or NLP evaluation [164]. Neural machine translation models are also very sensitive to input noise, such as character-level transformation [26, 146]. Finally, a few studies have explored adversarial examples for speech recognition [51, 76] and other speech processing tasks [190]; see [127] for an overview.

Other methods Erasure is an interesting approach to study neural networks for language, where certain components are erased or masked from the network [204]. These may be word embedding dimensions, hidden units, or even full words. The effect of erasure has been evaluated on word-, sentence-, and document-level tasks.

Several studies have conducted behavioral experiments to interpret word embeddings. A common formulation is to define an intrusion task, where a human is asked to identify an intruder word, chosen based on a difference in word embedding dimensions [113, 249].[10]

Since neural networks generate representations in vector space, a common approach is to find their nearest neighbors and observe them for qualitative trends, for example to analyze morphological and semantic similarity of words [177, 326, 330].

[10] The methodology follows earlier work on the interpretability of probabilistic topic models [60].

1.4 Language Representations

This thesis is focused on analyzing internal representations in deep learning models of language and speech processing. It is therefore useful to provide a brief overview of such representations. I will first focus here on the basic unit of a word, and then comment on representations of smaller and larger units.

You shall know a word by the company it keeps
— J. R. Firth

Word vector representations have been used in human language technology at least since the 1980s. However, they gained renewed popularity in recent years due to advances in developing efficient methods for inducing high quality representations from large amounts of raw text. A survey of different representations is given in [324], where three types of word representations are discussed. *Distributional* representations are based on co-occurrences statistics of words in some context, based on the *distributional hypothesis* that words appearing in similar contexts have similar meanings. Since these representations have dimensionality the size of the vocabulary, different dimensionality reduction techniques can later be applied. For example, using singular value decomposition (SVD) leads to latent semantic analysis (LSA) [96]. Another type of representation is based on *word clustering*, where Brown clustering is a notable example [46]. Finally, *distributed* word representations, also known as *word embeddings*, are low dimensional, real valued, dense vectors, where each dimension is a latent feature of the word.

Another way to categorize word vector representations is into *count* and *predict* models. The former type corresponds to the distributional representations and is based on co-occurrence counts. The latter corresponds to distributed representations (embeddings) and is based on predicting words in some context. Comparing these two types shows that distributed predictive word embeddings are superior in a variety of semantic tasks [22].[11]

[11] Interestingly, some prediction-based models can be cast as count-based models [198] so the distinction may not be that clear-cut.

Traditionally, distributed representations have been created by using neural network language models. A major obstacle in using such representations is that the neural language models are typically slow to train. Thus much work has focused on efficient methods for training such models [34, 77, 241]. An influential work has been the `word2vec` toolkit [237, 238], which implemented several successful algorithms for training distributed word representations. This work has led to many applications and extensions. For example, these embeddings were used in areas as diverse as sentiment analysis [8, 92], information retrieval [242], metaphor recognition [244], factoid [163] and community question answering [31, 246], summarization [169], semantic parsing [35], machine translation [354], dependency parsing [21, 62], and Chinese word segmentation [273].

While words are an important unit of human language, they do not tell the whole story. On the one hand, words combine to form larger meaningful units such as phrases, sentences, and passages. On the other hand, words are made of smaller units such as morphemes, phonemes, letters, and characters. We call the units above word level *super-word* elements and the ones below word level *sub-word* elements. Figure 1-3 illustrates this spectrum. Units above the word level tend to carry more complex semantic content, but a given super-word element (e.g., a sentence) does not recur very frequently. Sub-word units, on the other hand, have less semantic content but occur much more often.

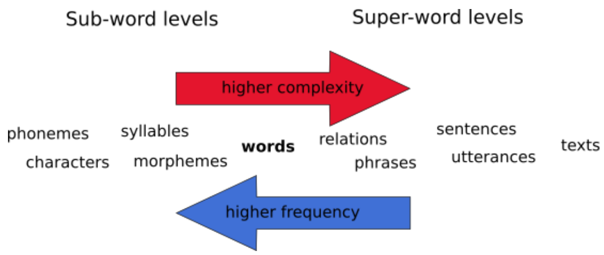
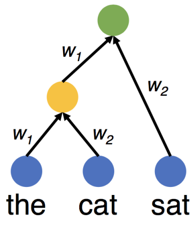


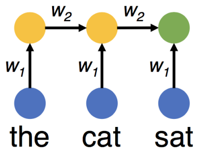
Figure 1-3: The spectrum of linguistic elements. Super-word units carry more complex semantic content but are less frequent. Sub-word units have less semantic content but occur more frequently.

Similarly to word vectors, document-level vector representations have also been around for a while, for example in the form of count-based bag-of-words (BOW) representations such as LSA or topic models based on latent Dirichlet allocation (LDA) [40]. With the rise of distributed vector representations for words there has been recent interest in finding analogous representations for both super-word and sub-word elements. In the former case, word vectors may be *composed* in different ways to obtain vector representations for phrases, sentences, and whole texts. For instance, a simple average composition is defined by an element-wise average of the word vectors. While this method ignores the word order in the text, it can lead to quite useful generic representations for texts of any length, and so it is a common baseline to compare with.[12]

More sophisticated methods for combining word vectors can be broadly classified into three types, according to the neural network architecture they employ: recursive neural networks (RecNNs), RNNs, and convolutional neural networks (CNNs). In RecNNs, the composition takes place along a syntactic tree. The composition function can be a single-layer neural network or more sophisticated compositions, and different options have been explored in various tasks [30, 308–312]. There are also extensions to multiple sentences [163, 199, 200]. The main difficulty with RecNN methods is their reliance on a syntactic parse tree. Such a structure may not be available for every language or domain, or it might be of poor quality.[13] An alternative approach processes the sentence (or text) word-by-word with an RNN. The final representation can be the last hidden state of the RNN, or a pooling of all states. RNNs tend to have difficulties dealing with long sequences. One approach to mitigating these problems is to use gating mechanisms such as LSTM networks [149]. Another common method is to add an attention mechanism, where the model learns to associate weights with different words in the sentence. This



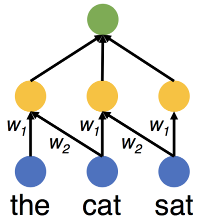
Recursive NN



Recurrent NN

[12] The average composition also turns out to capture basic sentence properties fairly well [2].
[13] It is also questionable whether trees are indeed needed for getting good sentence representations [202].

approach has been especially successful in machine translation [16, 221], as well as tasks that involve matching spans of text.[14] The machine translation models investigated in this work fall into this type of models. Finally, CNNs have been gaining popularity in a wide variety of tasks, including sentence classification [175], relation classification [93], machine translation [122], and other tasks [78]. A CNN captures local relationships between words through learned filter weights. Typically, a max-over-time pooling is performed to obtain a fixed-length vector representation for sentences of arbitrary length. Obviously, combinations of multiple architectures are possible, such as combining convolutional and recurrent networks for machine translation [121, 170, 194].[15]



Convolutional NN

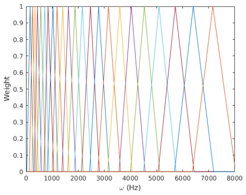
On the other end of the spectrum we find sub-word elements, starting with characters. Modeling the character level holds potential for alleviating common shortcomings of word representations. First, out-of-vocabulary (OOV) items can receive reasonable representations if the characters they are made of are similar to other in-vocabulary items. Second, character-based models are more robust to typos and non-standard forms that are ubiquitous in user generated content. Finally, character-based representations can generalize different morphological variants sharing the same basic concept, a common challenge in morphologically rich languages.[16] Vector representations of sub-word units have recently gained popularity in NLP. For example, character-based neural networks have been used in a number of tasks, including machine translation [219], language modeling [167, 177], POS tagging [208, 295], speech recognition [17, 222], dialect identification [28, 174], text classification [357], and factoid question answering [126]. Chapter 2 of this thesis investigates the use of character-based representations in neural machine translation.

[14] The leaderboards for the Stanford natural language inference (SNLI) [42] and question answering datasets (SQuAD) [281] demonstrate how important attention is in these tasks. In prior work, we found that attention also helps identify important text chunks in community question answering [286, 287].

[15] Recently, fully-attentional networks, with no explicit composition, have gained some popularity [327].

[16] Models based on characters also tend to have fewer parameters because they do not require a word embedding matrix which has the size of the vocabulary.

Before closing this section, a word on speech. Many representations of the speech signal have been considered in the context of speech recognition. Most of them start with Fourier analysis of the waveform, and compute a spectrogram showing the energy at each time/frequency point. Often, cepstral analysis is applied to de-convolve the source and the filter from the speech signal. The spectrum may first be transformed by Mel-scale filters that mimic human perception, placing a higher weight on energy in low-frequency regions. This results in the popular MFCCs. All these representations are covered in speech processing textbooks [151] and implemented in standard toolkits such as Kaldi [277].



Mel-spaced filters

In recent years, there have been attempts to move some or all of the speech signal processing into the neural network. For instance, in [290], a neural acoustic model with a time convolution learns better, and complementary, features compared to MFCCs. Others have investigated learning from the raw waveform and also found benefit in combining multiple representations as input to the neural network [325]. In Chapter 5, I study representations learned by an end-to-end ASR model that uses spectrogram features as input.

Finally, ideas from semantic text embeddings have started propagating to the speech domain, where researchers seek speech representations that capture the meaning of speech units such as words [64, 74, 75]. Other work found that grounding speech utterances in images helps obtain semantic representations of different units [5, 73, 141–143, 171].

1.5 Machine Translation

"Without translation, we would be living in provinces bordering on silence."
— George Steiner
"Translation is the art of failure."
— Umberto Eco
"Traduttore, traditore" — Italian proverb

1.5.1 Background

Recognizing ... the semantic difficulties because of multiple meanings, etc., I have wondered if it were unthinkable to design a computer which would translate. Even if it would translate only scientific material ... and even if it did produce an inelegant (but intelligible) result, it would seem to me worth while.
— Warren Weaver, Translation

Historical accounts of machine translation mention the 17th century as the time when initial thoughts of communicating between languages via mechanical devices (especially, mechanical dictionaries) first appeared.[17] However, these are best seen as early ideas of a "universal language", rather than machine translation [156]. Apart from interesting but mostly unnoticed patents for mechanical dictionaries in 1933 [156], the first proposal for a translation system is attributed to Warren Weaver in the 1940s, soon after the invention of electronic computers [336]. Weaver's memorendum on Translation had widespread influence and "launched machine translation as a scientific enterprise in the United States and subsequently elsewhere" [156]. In it, Weaver outlined four strategies for machine translation: determining the meaning of a word from its context, formal proofs that translation by a computer is logically possible, a cryptographic view of translation in light of Shannon's probabilistic communication theory, and language universals. Many of these ideas have been picked up by the nascent machine translation community in subsequent years [156].

[17] The brief history outlined here is based on several accounts of machine translation history [156–158, 181, 182, 227, 338]. The Machine Translation Archive also contains many informative sources: `http://www.mt-archive.info`.

The excitement of initial years had been faced with limited success in producing high-quality and scalable machine translation systems. While development of systems operating in closed domains continued, much of the funding – and research – on machine translation had been cut in the 1960s [227].[18]

> *... one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'*
> — Warren Weaver, Translation

Research has continued through the 1970s and 1980s with interlingual and transfer ideas [156, 227], until the statistical revolution of the 1990s. Most influential were the IBM statistical models [44, 45] that adapted prior work on speech recognition to the translation case [19]. Inspired by Weaver's cryptographic and statistical ideas, these models estimated the probability of translating source sentence s to target sentence t with Bayes' theorem in a noisy channel model:

$$P(t|s) = \frac{P(t)P(s|t)}{P(s)} \tag{1.3}$$

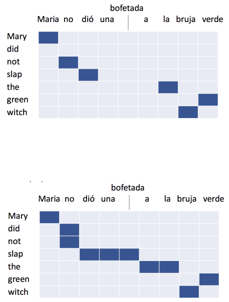
Since the goal is to maximize $P(t|s)$, this results in what is known as *the fundamental equation of machine translation*:

$$\hat{t} = \arg\max_{t} P(t)P(s|t) \tag{1.4}$$

The IBM models defined a probability model with two components: the translation model $P(s|t)$ and the language model $P(t)$. The language model can be easily estimated from large amounts on raw texts, for example with n-gram language models [181]. For the translation model, the IBM papers have introduced a series of models relying on word alignments, which are estimated from parallel sentences using expectation maximization

[18] A 1960 report by Bar-Hillel was influential in determining that "Fully automatic, high quality translation is not a reasonable goal, not even for scientific texts". With the 1966 ALPAC (Automatic Language Processing Advisory Committee) report, funding for machine translation saw massive cuts [227]. See [156] for an interesting, and critical, discussion of this period.

(EM) algorithms [44, 181, 227].



Example word and phrase alignments, adapted from [227]

The statistical paradigm took over, and publicly-available tools for word alignment and statistical machine translations have become increasingly popular. In the 2000s, phrase-based approaches [185, 262] have proved very successful and became widely used [227]. The Moses phrase-based system has been particularly popular [186].

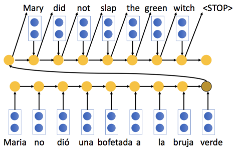
Along with phrase-based statistical machine translation, other work has explored hierarchical phrases [65, 66] and more linguistically motivated approaches to syntax-based statistical machine translation [338] such as mapping between syntactic trees and linear word sequences in the source and/or target language [115, 213, 216, 305, 347, 348]. Work on phrase-based and syntax-based statistical machine translation continued into the 2010s, until the revival and then takeover of neural machine translation.

1.5.2 Neural Machine Translation

Early work involving neural networks in machine translation includes [331], where a neural parser was integrated in a speech translation system, as well as more independent neural approaches for machine translation [54, 72, 109, 159, 231, 334]. These, however, were very limited in scale.[19] In the late 2000s, neural networks started showing benefit in full-scale machine translation systems. Initially, they were incorporated in certain parts of previous statistical machine translation systems, such as language models [89, 298, 300], ordering models [86, 172, 206], or other components [218, 299]. The first successful large-scale approaches to end-to-end neural machine translation used convolutional [170] and recurrent neural networks [68, 318]. The sequence-to-sequence framework in [318] was particularly influential. Their model is made of two neural networks: an encoder and a decoder. The encoder maps a source sentence to a vector representation, which the decoder

[19] As noted by Koehn [182], some of these models are remarkably similar to the modern encoder-decoder approach to neural machine translation; see for example figure 1 in [109].

then maps to the target translation. The two modules are optimized jointly such that the model can be trained end-to-end with gradient descent on example translations.



Encoder-decoder neural machine translation

More formally, given a source sentence $s = \{w_1, w_2, \ldots, w_N\}$ and a target sentence $t = \{u_1, u_2, \ldots, u_M\}$, the model first generates a vector representation for the source sentence using an encoder (Equation 1.5) and then maps this vector to the target sentence using a decoder (Equation 1.6):

$$\texttt{ENC} : s = \{w_1, w_2, \ldots, w_N\} \mapsto \boldsymbol{s} \in \mathbb{R}^k \tag{1.5}$$

$$\texttt{DEC} : \boldsymbol{s} \in \mathbb{R}^k \mapsto t = \{u_1, u_2, \ldots, u_M\} \tag{1.6}$$

The encoder-decoder model is trained jointly on a corpus of example translations $\{s^{(i)}, t^{(i)}\}$ by maximizing the log-likelihood of the data:

$$\sum_i \sum_{j=1}^{|t^{(i)}|} \log P(u_j^{(i)} | u_1^{(i)}, \ldots, u_{j-1}^{(i)}, s^{i}) \tag{1.7}$$

The encoding and decoding steps assume a vector representation $\boldsymbol{w} \in \mathbb{R}^d$ for each word in the vocabulary. Typically, the encoder and decoder are modeled as RNNs, such as LSTM [149] or gated recurrent unit (GRU) networks [69]. The encoder takes the current word vector $\boldsymbol{w}_t$ and the previous source hidden state $\boldsymbol{h}_{t-1}^S$, and computes a hidden state recursively: $\boldsymbol{h}_t^S = \texttt{ENC}(\boldsymbol{h}_{t-1}^S, \boldsymbol{w}_t)$. The decoder similarly computes the hidden states on the target side: $\boldsymbol{h}_t^T = \texttt{DEC}(\boldsymbol{h}_{t-1}^T, \boldsymbol{u}_t)$. Then, the decoder predicts the next target word by mapping the hidden state to the vocabulary size V, $\boldsymbol{y}_t = \boldsymbol{W}^{yh}\boldsymbol{h}_t \in \mathbb{R}^V$, and computing a Softmax: $P(u_{t+1} = k | \boldsymbol{y}_t) = \frac{\exp(y_{t,k})}{\sum_{k'=1}^{V} \exp(y_{t,k'})}$.[20]

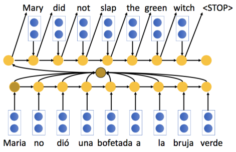
Two more improvements are needed for obtaining a state-of-the-art neural machine

[20] Refer to [129, 318] for more details.

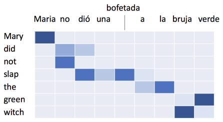
translation system. First, it is common to stack multiple layers [16, 133, 318],[21] such that the encoder hidden state at layer l is conditioned on layer $l-1$ (Equation 1.8).[22]

$$\boldsymbol{h}_t^{S,l} = \texttt{ENC}(\boldsymbol{h}_{t-1}^{S}, \boldsymbol{h}_t^{S,l-1}, \boldsymbol{w}_t) \tag{1.8}$$

The second improvement concerns the conditioning of the decoder on the source sentence encoding. In the above sequence-to-sequence formulation, the source sentence has one fixed representation, the last encoding hidden state $\boldsymbol{h}_N^S$, which is used to initialize the decoder's hidden state and thus conditions the decoder. This means that information from the encoder is more salient during the initial decoding steps, but then becomes less accessible as decoding proceeds. It also enforces a strong assumption that all information relevant for decoding needs to be captured in one vector representation. An alternative is to use all of the encoder's hidden states by weighting their contribution to each decoding step [16]. This so-called attention mechanism allows the decoder to attend to different source states during decoding. The attention weights are parameterized and conditioned on previous decoding decisions, forming a soft alignment between source and target words.[23]



Attention-based encoder-decoder neural machine translation



Attention alignment

Recent developments The field of neural machine translation is moving fast and new improvements appear very frequently. The models studied in this thesis are standard encoder-decoder models with attention, based on RNNs. More recent developments include fully-convolutional models [122], purely attention-based models [327], and even non-autoregressive models [135]. While the final word about the best architecture has not yet been spoken, the models studied in this work remain highly influential and are basic models that are implemented in all major neural machine translation toolkits.

[21] Typical numbers are 2–4 layers, although deeper models have also been considered [43, 344, 361].

[22] Stacking is usually done by feeding the output of each layer to the input of the layer above it [133], although other options have been explored [235].

[23] The specific kind of attention used here is global-general-attention with input-feeding [221].

1.6 Speech Recognition

> *"Every field has its Holy Grail, and automatic speech recognition (ASR) is ours."*
> — James L. Flanagan

1.6.1 Background

The first system for the automatic recognition of speech is attributed to a digit recognizer developed at Bell Labs [85] that measured spectral energy in two wide bands, approximating the first and second formants.[24] It achieved 97–99% accuracy on recognizing digits from a single speaker.[25] Similar systems expanded the number of recognized sounds and in 1959 statistical information regarding phoneme transition probabilities was first used [88, 111].

The late 1960s and 1970s saw breakthroughs along several lines. New methods for feature extraction were developed, namely, the fast Fourier transform (FFT), cepstral analysis, and linear predictive coding (LPC) [125, 227]. Pattern matching algorithms were developed and applied to speech processing: the deterministic dynamic time warping (DTW) and the probabilistic hidden Markov model (HMM). Template-based matching of isolated words was the standard [112, 125], although there were also attempts at continuous speech recognition [284]. At the same time, automatic speech recognition systems started handling medium vocabularies (thousands of words). DARPA programs were instrumental in promoting speech recognition research in multiple laboratories.

[24] This section is largely based on the historical accounts in [112, 125, 227].

[25] A dog toy named "Radio Rex" is sometimes mentioned as an earlier recognizer [125, 227]. It had a spring that was released by 500 Hz acoustic energy, roughly corresponding to the first formant of the vowel /eh/ in "Rex".

In the 1980s, continuous speech recognition became common and vocabulary sizes increased to up to 60,000 words [227]. Larger speech corpora were collected, such as the TIMIT acoustic-phonetic dataset [120], the most popular LDC corpus. The template matching approach was replaced by a statistical modeling approach [19]. According to a noisy channel model, given a speech input $\boldsymbol{x}$ and its transcribed label sequence l, the probability $P(l|\boldsymbol{x})$ is can be written using Bayes' theorem as:

$$P(l|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|l)P(l)}{P(\boldsymbol{x})} \tag{1.9}$$

The model seeks to maximize the probability $P(l|\boldsymbol{x})$:

$$\hat{l} = \arg\max_{l} P(l)P(\boldsymbol{x}|l) \tag{1.10}$$

The probability $P(l)$ is called the language model and can be estimated from raw texts. The probability $P(\boldsymbol{x}|l)$ is the acoustic model, which may be estimated by a Gaussian mixture model (GMM). The acoustic model and the language model are combined using a decoding algorithm such as Viterbi decoding [152].

This formulation has been the predominant one for several decades, with many subsequent improvements [125].[26] Work has also shifted to larger vocabularies and more challenging scenarios like conversational speech [125]. Toolkits for ASR appeared in the 1990s and 2000s. Recently, Kaldi has been particularly popular [277].

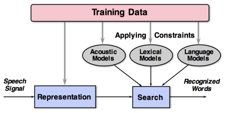
Neural networks have been considered from time to time in work on automatic speech recognition. Digit recognizers using neural networks were implemented already in the 1960s [125]. In the 1980s, several systems made use of neural networks for phoneme recognition [332] or vowel and consonant classification [211, 224]. Hybrid approaches

[26] Examples include finite-state methods [245], discriminative training [71, 276], segment-based methods [124, 363], and a variety of language models [152].

combining HMMs with neural networks also appeared [125]. Interest in neural networks rose again in the late 2000s with hybrid acoustic modeling approaches showing promising results, this time with deep neural networks.[27] RNNs, especially LSTMs, were particularly successful, first in phone recognition on TIMIT [132, 133] and then in larger tasks [292]. Combinations of multiple network types, such as stacking CNNs, RNNs and fully-connected layers, were also successful [289].

Another important neural ingredient in ASR is in the language model, where neural language models have provided significant gains [236].

1.6.2 End-to-End Speech Recognition



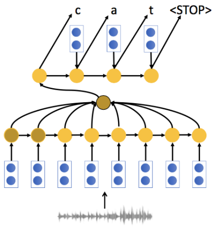
Traditional ASR pipeline (MIT 6.345 class notes)

Traditional automatic speech recognition (ASR) systems are composed of multiple components, including an acoustic model, a language model, a lexicon, and possibly other components. Each of these is trained independently and combined during decoding. As such, the system is not directly trained on the speech recognition task from start to end. In contrast, end-to-end ASR systems aim to map acoustic features directly to text (words or characters). Such models have recently become popular in the ASR community thanks to their simple and elegant architecture [18, 59, 70, 130, 222, 234]. Recent advances in end-to-end ASR also achieve impressive performance [67, 359, 360].

There are two main paradigms in end-to-end ASR: connectionist temporal classification (CTC) [9, 107, 130, 234] and attention-based sequence-to-sequence (seq2seq) models [18, 59, 70]. The seq2seq approach first encodes the sequence of acoustic features into a single vector and then decodes that vector into the sequence of symbols (characters). Formally, let $\boldsymbol{x} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ denote sequence of acoustic features[28] and let $l = (l_1, \dots, l_M)$ denote its transcription (for example, a sequence of characters or words).

[27] The review in [351] provides many useful references on architectures and training.

[28] For example, MFCCs, spectrograms of frequency magnitudes, or even raw waveform.



Attention-based seq2seq ASR

An encoder generates a vector representation for the utterance (Equation 1.11), which a decoder then maps to the label sequence (Equation 1.12):

$$\texttt{ENC} : \boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \mapsto \boldsymbol{u} \in \mathbb{R}^k \tag{1.11}$$

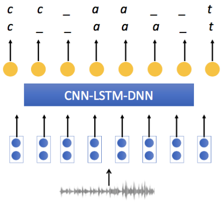
$$\texttt{DEC} : \boldsymbol{u} \in \mathbb{R}^k \mapsto l = \{l_1, \ldots, l_M\} \tag{1.12}$$

The encoder-decoder model is trained jointly on a corpus of utterances and their transcriptions, $\{x^{(i)}, l^{(i)}\}$, by maximizing the log-likelihood of the data:

$$\sum_i \sum_{j=1}^{|l^{(i)}|} \log P(l_j^{(i)} | l_1^{(i)}, \ldots, l_{j-1}^{(i)}, \boldsymbol{x}^{i)}) \tag{1.13}$$

As in neural machine translation, the attention mechanism improves upon this method by conditioning on a different summary of the input sequence at each decoding step [59, 70].

An alternative approach to end-to-end ASR is based on CTC, which avoids the need to condense the full utterance into one vector representation. The CTC model is based on an RNN that takes acoustic features as input and predicts one symbol per each frame. Symbols are typically characters, in addition to a special blank symbol. The CTC objective function [131] marginalizes over all possible sequences of symbols given a transcription:



CTC-based ASR

$$\sum_i \log p(l^{(i)} | \boldsymbol{x}^{(i)}) \tag{1.14}$$

where the probability of a label sequence l given an input sequence $\boldsymbol{x}$ is defined as:

$$p(l|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(l)} p(\pi|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(l)} \prod_{j=1}^{|l^{(i)}|} \phi_j^K(\boldsymbol{x})[\pi_j] \tag{1.15}$$

where $\mathcal{B}$ removes blanks and repeated symbols, $\mathcal{B}^{-1}$ is its inverse image, and $\phi_j^K(\boldsymbol{x})[r]$ is unit r of the model output after the top Softmax layer at time j, which is interpreted as the probability of observing label r at time j. This formulation allows mapping long frame sequences to short character sequences by marginalizing over all possible sequences containing blanks and duplicates.

The ASR model may be a deep model, with K layers, where $\phi_j^k(\boldsymbol{x})$ represents the output of layer k at time j. Layer K is the Softmax layer, which maps to the label size (for example, the size of the alphabet plus the blank symbol).

Both of these approaches to end-to-end ASR usually predict a sequence of characters, although there have also been initial attempts at directly predicting words [13, 313].

1.7 Summary of Contributions

This thesis lays down a methodological approach for studying internal representations in end-to-end deep learning models. The methodology has three main steps: *(i)* training an end-to-end model on a complex task; *(ii)* generating internal feature representations with the trained model for a simpler task; and *(iii)* training and evaluating a classifier on the simpler task. This process provides a quantitative evaluation of the representations for a given task of interest.

The methodology is tested on two important human language technology problems—machine translation and speech recognition—by evaluating a variety of simple tasks that target linguistic properties. Specifically, I study neural machine translation from the perspective of POS tagging, morphological tagging, semantic tagging, syntactic dependency labeling, and semantic dependency labeling. I also investigate speech recognition via a frame-level phonetic classification task.

The analysis of the results yields interesting insights regarding representation learning in end-to-end deep learning models. The main important insights are:

- Deep neural networks that are trained in an end-to-end fashion learn a non-trivial amount of linguistic information without being provided with direct supervision during the initial training process.

- Linguistic information tends to be organized in a modular manner, whereby different parts of the neural network generate representations with varying amounts and types of linguistic properties.

- In particular, a hierarchy of language representations emerges in networks trained on the complex tasks studied in this thesis. In the machine translation case, lower layers of the network focus on local, low-level linguistic properties (morphology, POS, local relations), while higher layers are more concerned with global, high-level properties (lexical semantics, long-range relations). In the speech recognition case, phonetic information is better captured in intermediate layers of the network, while the top layers are more tuned to predicting character sequences.

- The encoder and decoder in sequence-to-sequence neural machine translation both capture a significant amount of morphological information. Nevertheless, injecting morphological knowledge into the decoder leads to improved representations and better performance on the translation task.

- Differences in architecture correspond to different qualities of language representations. For instance, networks with access to character information generate representations that contain more morphological information than purely word-level networks. This is especially important for representing infrequent words.

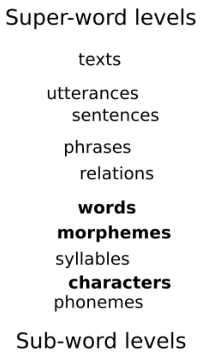
Chapter 2

Word Structure and Neural Machine Translation: Morphology

"I goed", "I clomb",
"I'm becarefulling"
— A 3-year-old learning morphology

2.1 Introduction

Capturing morphology, or word structure, is an important problem in machine translation. Languages with rich morphological systems exhibit a large number of surface forms for each lemma. This poses problems of data sparsity, as many word forms will not be seen frequently enough in the training data for correctly translating them [181]. Therefore, machine translation systems resort to different techniques when handling morphologically-rich languages. First, morphological segmentation can reduce sparsity by sharing of information between words with similar stems or other morphemes. Such segmentation has been shown to improve machine translation performance [6, 15, 139, 184, 258]. Word segmentation may also be helpful even when it does not strictly correspond to meaningful units (morphemes), as shown by unsupervised methods for obtaining sub-word units [108, 304, 315, 329, 344].[1] Another method for handling morphology in machine translation is to use various morphological properties as features, an approach that has been extensively studied in non-neural machine translation [98, 110, 134, 154, 183, 240, 321], and more recently in neural machine translation [153, 302]. Lastly, neural machine translation facilitates the use of character-aware representations, where a word may be represented as a sequence of characters that is processed in a sub-network [27, 82, 209, 219, 291, 330]. Such models maintain the notion of a word, but perform hierarchical processing from characters, through words, to sentences.[2] More extreme approaches dispense with the notion of a word and view the entire sentence as just a sequence of characters [194, 349], although the space character may serve as an implicit word boundary marker.



Using character-aware representations is attractive for several reasons. It does not require any pre-processing or post-processing and can be trained in an end-to-end manner. Using characters may alleviate the high computation load entailed by word representations when the word vocabulary is large. And importantly, the representations contain character information that may be helpful for capturing typos and misspellings, as well as morphological properties.

This chapter investigates what kind of morphological information is captured by neural machine translation models. The linguistic units of study are words and their sub-parts, characters and morphemes. The work here aims to provide quantitative, data-driven an-

[1] Whether unsupervised word segmentation works as well as supervised morphological segmentation in neural machine translation is an open question and may well be language-specific, as recent studies have produced conflicting results [153, 291].

[2] An earlier approach combining word and sub-word units for statistical machine translation is found in [220].

swers to the following questions:

1. Which parts of the neural machine translation architecture capture word structure?

2. What is the division of labor between different components of the network? For example, is morphology better represented in different layers of the network? What about representations of source and target languages in the encoder and decoder networks, respectively?

3. How do different word representations help learn better morphology and modeling of infrequent words? Do models with access to characters learn representations that are more informative for morphology?

4. How does the target language affect the learning of word structure? Does translating into different languages requires learning different source-side representations?

To answer such questions, I focus on the tasks of part-of-speech (POS) and full morphological tagging, which is the identification of all pertinent morphological features for every word. I define word-level classification tasks, where representations from different parts of the neural machine translation model are used for predicting these properties. I investigate how different systems capture POS and morphology through a series of experiments along several parameters. For instance, I contrast word-based and character-based representations, use different encoding layers, vary source and target languages, and compare extracting features from the encoder vs. the decoder.

The experiments employ several languages with varying degrees of morphological richness: French, German, Czech, Arabic, and Hebrew. They reveal interesting insights such as:

- Character-based representations are much better for learning morphology, especially for low-frequency words. This improvement is correlated with better translation performance. On the other hand, word-based models are sufficient for learning the structure of common words.

- Lower layers of the encoder are better at capturing word structure, while deeper networks improve translation quality. This suggests that higher layers focus more on word meaning, an idea we will return to in Chapter 3.
- The target language impacts the kind of information learned by the machine translation system. Translating into morphologically-poorer languages leads to better source-side word representations. This is partly, but not completely, correlated with translation quality.
- The neural encoder and decoder learn representations of similar quality. The attention mechanism affects the quality of the encoder representations more than that of the decoder representations. Section 2.7 explores how to improve the neural machine translation system by injecting morphological knowledge to the decoder.

2.2 Related Work

Machine translation systems that deal with morphologically-rich languages resort to various techniques for representing morphological knowledge, such as word segmentation [15, 184, 258] and factored translation and reordering models [98, 183]; see [181] for an overview. Characters and other sub-word units have become increasingly popular in neural machine translation, although they had also been used in phrase-based MT for handling morphologically-rich [220] or closely related language pairs [97, 253]. In neural machine translation, such units are obtained in a pre-processing step—for example, with byte-pair encoding [304] or the word-piece model [344]—or learned during training with a character-based convolutional or recurrent sub-network [82, 219, 330]. The latter approach has the advantage of maintaining the original word boundaries without requiring pre- and post-processing. Relatedly, I explore a character convolutional neural network (CNN) which has been used in language modeling and machine translation [27, 82, 167, 177, 291], evaluate the quality of different representations learned by a system augmented with this subnetwork in terms of POS and morphological tagging, and contrast them with a purely word-based system.

There is little prior work on analyzing neural machine translation from the perspective of morphology. A relevant work is [330], which analyzes different representations for morphologically-rich languages in neural machine translation, but does not directly measure the quality of the learned representations.

2.3 Methodology

The methodological approach taken here for studying morphological information in neural machine translation is an instantiation of the high-level approach presented in Section 1.2. It is based on the following three steps: *(i)* train a neural MT system on a parallel corpus; *(ii)* use the trained model to generate feature representations for words in a language of interest; and *(iii)* train a classifier using generated features to make predictions for a morphology prediction task. The quality of the trained classifier on the given task serves as a proxy to the quality of the extracted representations. It thus provides a quantitative measure of how well the original MT system learns features that are relevant to the given task. Figure 2-1 illustrates this process for the neural machine translation encoder. A similar procedure is used for for analyzing representations in the decoder.

The translation model used in the following experiments is a 2-layer long short-term memory (LSTM) encoder-decoder with attention (Section 1.5.2). The model is trained using a standard implementation [176] with the following default settings: word vectors

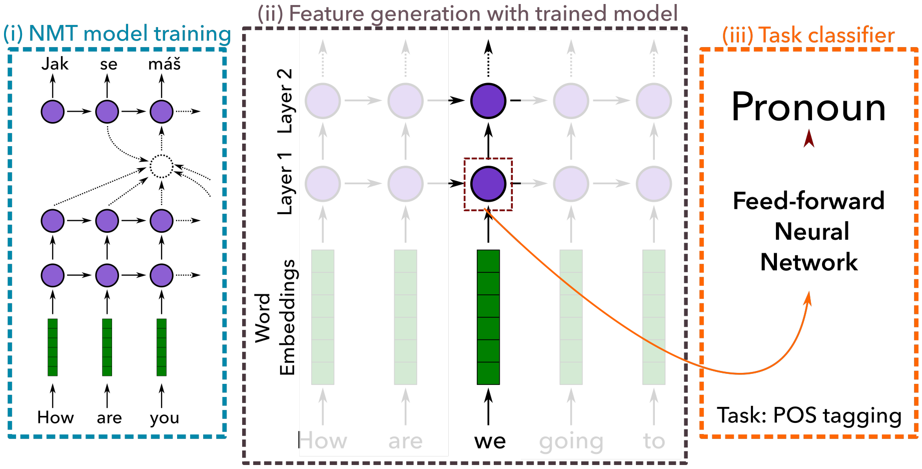


Figure 2-1: Methodology for analyzing morphology in neural machine translation representations. *(i)* a neural machine translation system is trained on a parallel corpus; *(ii)* the trained model is used for generating features; *(iii)* a classifier is trained using the generated features. In this case, a POS tagging classifier is trained on features from the first hidden layer in the encoder.

and LSTM states have 500 dimensions, stochastic gradient descent (SGD) with initial learning rate of 1.0 and rate decay of 0.5, and dropout rate of 0.3. The character-based model is a CNN with a highway network over characters [177] with 1000 feature maps and a kernel width of 6 characters. This model was found to be useful for translating morphologically-rich languages [27, 82]. The machine translation system is trained for 20 epochs, and the model with the best loss on the development set is used for generating features for the classifier.

The classifier is modeled as a simple feed-forward neural network with one hidden layer, dropout ($\rho = 0.5$), a rectified linear unit (ReLU) non-linearity, and an output layer mapping to the tag set (followed by a Softmax). The size of the hidden layer is set to be identical to the size of the encoder/decoder's hidden state (typically 500 dimensions). The objective function is cross-entropy, optimized by Adam [179] with the recommended parameters ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = e^{-8}$). Training is run with shuffled mini-batches of size 16 and stopped once the loss on the development set stops improving (allowing a patience of 5 epochs).

A note on the choice of classifier. If the goal is to obtain the best results on predicting morphology, then a powerful classifier might be desirable (for instance, an LSTM over encoder states). However, using a non-contextual classifier enables focusing on the quality of the representations learned by the machine translation system rather than obtaining state-of-the-art morphological prediction performance. Arguably, if the learned representations are good, then a non-linear classifier should be able to extract useful information from them.[3]

2.4 Data

The experiments on morphology prediction are conducted with several language pairs, including morphologically-rich languages, that have received relatively significant attention in the machine translation community: Arabic-English, German-English, French-English, and Czech-English. Additional experiments broaden the analysis by studying Arabic-Hebrew, two languages with rich and similar morphological systems, and Arabic-German, two languages with rich but different morphologies.

MT data The dataset used for training machine translation models is the WIT^3 corpus of TED talks [55, 56] made available for IWSLT 2016. This allows for comparable and cross-linguistic analysis. Statistics about each language pair are given in Table 2.1 (under Predicted).[4] The official development and test sets are used for tuning and testing.[5] In the

[3] Note that in a few controlled experiments, a linear classifier produced similar trends to the non-linear one, but overall lower results; Qian et al. [279] reported similar findings.

[4] The datasets and more statistics are available at `https://wit3.fbk.eu`.

[5] For Arabic-Hebrew, the experiments follow the split in [27]

	Arabic		German		French	Czech
	Gold	Predicted	Gold	Predicted	Predicted	Predicted
Train Tokens	0.5M	3.7M	0.9M	4M	5.2M	2M
Dev Tokens	63K	41K	45K	50K	55K	35K
Test Tokens	62K	79K	44K	25K	23K	20K
Train Sentences	16K	0.2M	47K	0.2M	0.2M	0.1M
Dev Sentences	1984	2456	1500	2452	2551	1991
Test Sentences	1950	5177	1500	5431	4273	4223
POS Tags	42		54		33	368
Morphological Tags	1969		214		–	–

Table 2.1: Statistics for annotated corpora used in morphology prediction experiments, with either gold or predicted tags. The numbers with predicted tags correspond to the non-English side in Arabic/German/French/Czech-English parallel data.

experiments below, the reported results are averages over test sets.

Annotated data Two kinds of datasets were used for training POS and morphological classifiers: gold-standard and predicted tags. The predicted tags were obtained by annotating the parallel data with freely available taggers, while gold tags are extracted from human-annotated datasets.[6] Table 2.1 provides statistics for datasets with gold and predicted tags. The classifiers were trained on predicted annotations, and similarly on gold annotations, when these are available.

Experiments using gold tags were conducted on the Arabic Treebank for Arabic[7] and the Tiger corpus for German.[8] The following tools were used to annotate the parallel corpora: MADAMIRA [272] for Arabic POS and morphological tags, Tree-Tagger [296] for Czech and French POS tags, LoPar [297] for German POS and morphological tags, and

[6] Using predicted tags is necessary when studying representations on the decoder side. For fair comparison, results are also reported with predicted tags on the source side.

[7] Experiments followed the versions and splits described in the MADAMIRA manual [272].

[8] Two sets with 1500 sentences each were randomly chosen for development and test, since the Tiger corpus does not have a specified split. The Tiger corpus is available at `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`.

MXPOST [283] for English POS tags.[9] As mentioned before, our goal is not to achieve state-of-the-art results, but rather to study what different components of the neural machine translation architecture learn about word morphology.

2.5 Encoder Analysis

The encoder processes the source sentence and produces a vector representation for word t at layer l, $\boldsymbol{h}_t^{S,l}$ (Equation 1.8, repeated here):

$$\boldsymbol{h}_t^{S,l} = \texttt{ENC}(\boldsymbol{h}_{t-1}^{S}, \boldsymbol{h}_t^{S,l-1}, \boldsymbol{w}_t) \tag{1.8}$$

This section studies the impact of several aspects on the quality of these representations from the perspective of morphology: character-based vs. word-based representations, what happens at different layers in the encoder, and how translating into different target languages affects the source-side representations.

2.5.1 Effect of word representation

Neural machine translation models tend to perform better when they have access to characters and other sub-units (Section 2.1). Do such models also learn better representations in terms of morphology? Table 2.2 shows POS and morphological tagging accuracy using features from different character-based and word-based encoders. Character-based models always generate better representations for POS tagging, especially in the case of morphologically-richer languages like Arabic and Czech. The superior morphological power of the char-based model also manifests in better translation quality (measured by

[9] These tools are recommended on the Moses website: `http://www.statmt.org/moses/?n=Moses.ExternalTools`

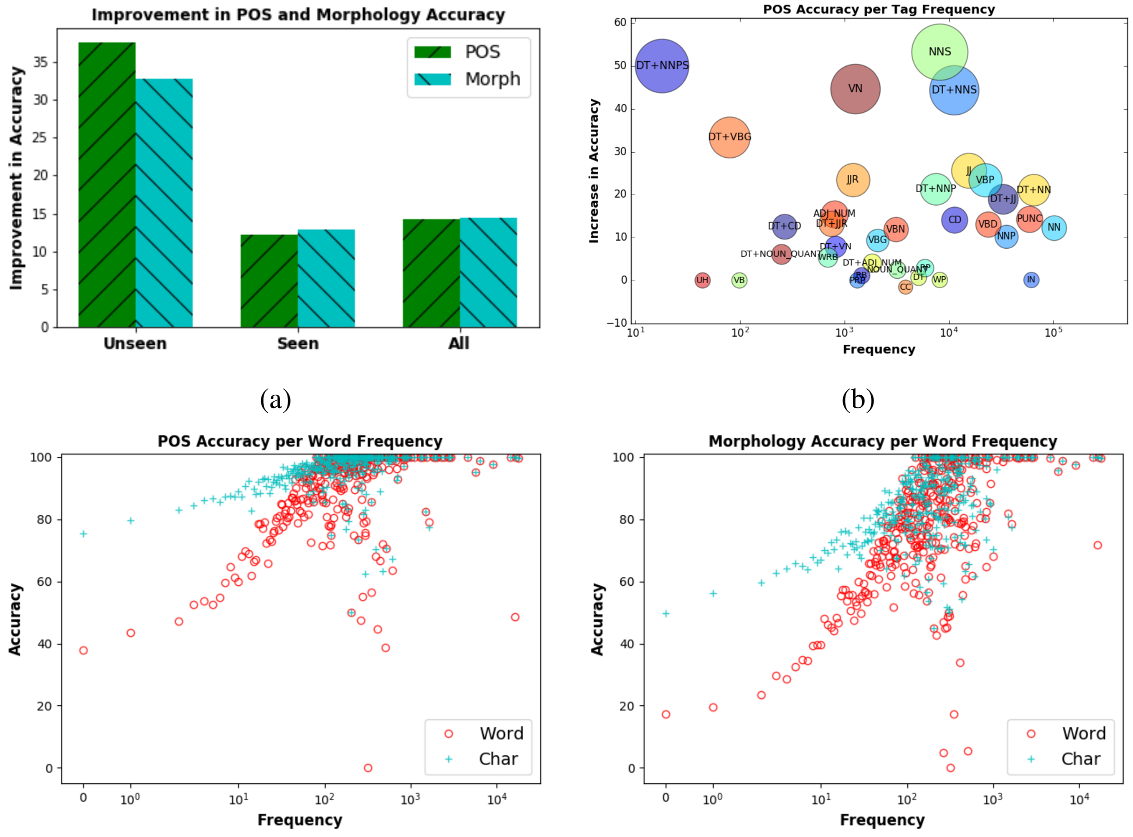
	POS		Morphology		
	Gold	Pred	Gold	Pred	BLEU
	Word/Char	Word/Char	Word/Char	Word/Char	Word/Char
Arabic-English	80.31/93.66	89.62/95.35	67.66/81.61	78.27/84.15	24.7/28.4
Arabic-Hebrew	78.20/92.48	88.33/94.66	65.20/79.66	77.55/83.51	9.9/10.7
German-English	87.68/94.57	93.54/94.63	–	88.67/90.45	29.6/30.4
French-English	–	94.61/95.55	–		37.8/38.8
Czech-English	–	75.71/79.10	–		23.2/25.4

Table 2.2: Effect of word representation on encoder representations: POS and morphological tagging accuracy on gold and predicted tags using word-based and character-based representations, as well as corresponding BLEU scores. Character-based representations always lead to better representations as well as higher BLEU scores.

BLEU [270]), as shown in the table.

The word-based representations are quite weak in the case of gold tags, which can be attributed to the change of domains: gold tags are on a different corpus than the translation corpus, while predicted tags are on the same corpus. This leads to a large degree of unknown words when training classifier on word-based vs. character-based representations. Let us examine the impact of word frequency more closely in an example case: Arabic POS and morphological tagging using gold tags. Figure 2-2a shows the effect of using word-based vs. character-based feature representations, obtained from the encoder of the Arabic-Hebrew system. Clearly, the character-based model is superior to the word-based one. This is true for the overall accuracy (+14.3% in POS, +14.5% in morphology), but even more so in out-of-vocabulary (OOV) words (+37.6% in POS, +32.7% in morphology). Figures 2-2c and 2-2d show that the gap between word-based and char-based representations increases as the frequency of the word in the training data decreases. In other words, the more frequent the word, the less need there is for character information. These findings make intuitive sense: the char-based model is able to learn character n-gram patterns that are important for identifying word structure, but as the word becomes more frequent the word-based model has seen enough examples to make a decision.

Figure 2-2b plots the difference in POS accuracy when moving from word-based to character-based representations, per frequency of the tag in the training data. Tags closer to the upper-right corner of the figure occur more frequently in the training set and are better predicted by character-based compared to word-based representations. There are a few fairly frequent tags (in the middle-bottom part of the figure) whose accuracy does not improve much when moving from word-based to character-based representations: mostly conjunctions, determiners, and certain particles (CC, DT, WP). But there are several very frequent tags (NN, DT+NN, DT+JJ, VBP, and even PUNC) whose accuracy improves quite a lot. Then there are plural nouns (NNS, DT+NNS) where the character-based model really shines. This makes sense linguistically as plurality in Arabic is usually expressed by certain suffixes ("-wn/yn" for masculine plural, "-At" for feminine plural). The character-based model is thus especially good with frequent tags and infrequent words, which is understandable given that infrequent words typically belong to frequent open categories like nouns and verbs.



(a)

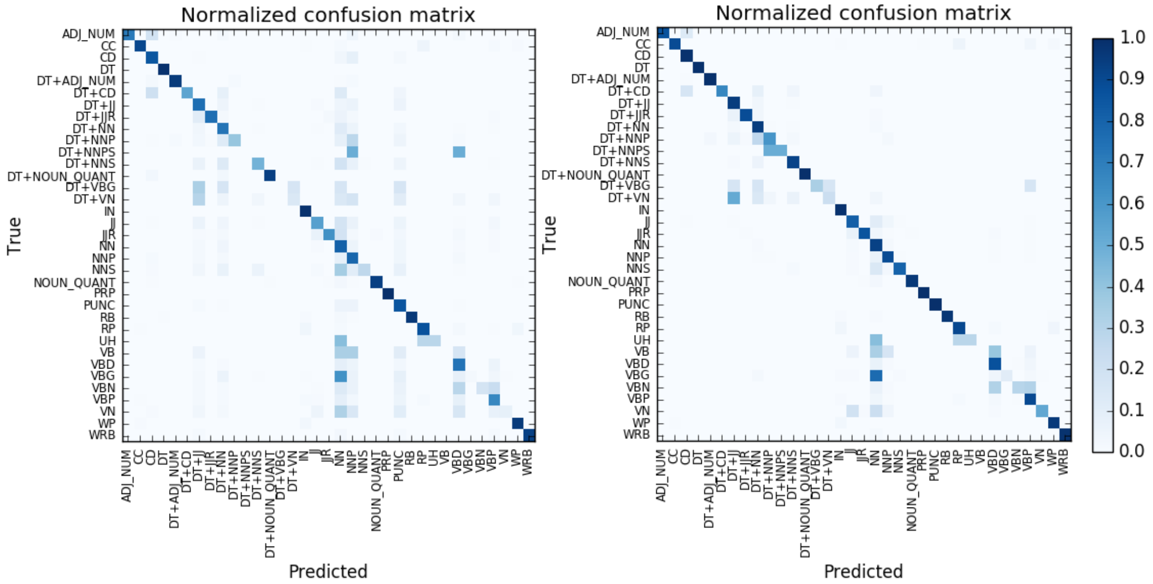
(b)

(c)

(d)

Figure 2-2: The effect of frequency on character-based and word-based representations. **(a)** Improvement in POS/morphology accuracy of character-based vs. word-based models for words unseen/seen in training, and for all words. **(b)** Increase in POS accuracy with character-based vs. word-based representations per tag frequency in the training set; larger bubbles reflect greater gaps. **(c/d)** POS/Morphology accuracy of word-based and character-based models per word frequency in the training data.

Figure 2-3 plots confusion matrices for POS tagging using word-based and character-based representations (from Arabic encoders). While the character-based representations are overall better, the two models still share similar misclassified tags. Much of the confusion comes from wrongly predicting nouns (`NN`, `NNP`). In the word-based case, relatively many POS tags with determiner (`DT+NNP`, `DT+NNPS`, `DT+NNS`, `DT+VBG`) are wrongly predicted as non-determined nouns (`NN`, `NNP`). In the character-based case, this hardly happens. This suggests that the character-based representations are predictive of the presence of a determiner, which in Arabic is expressed as the prefix "Al-"[10] (the definite article), a pattern easily captured by a character-based model.



(a) Word-based representations.

(b) Character-based representations.

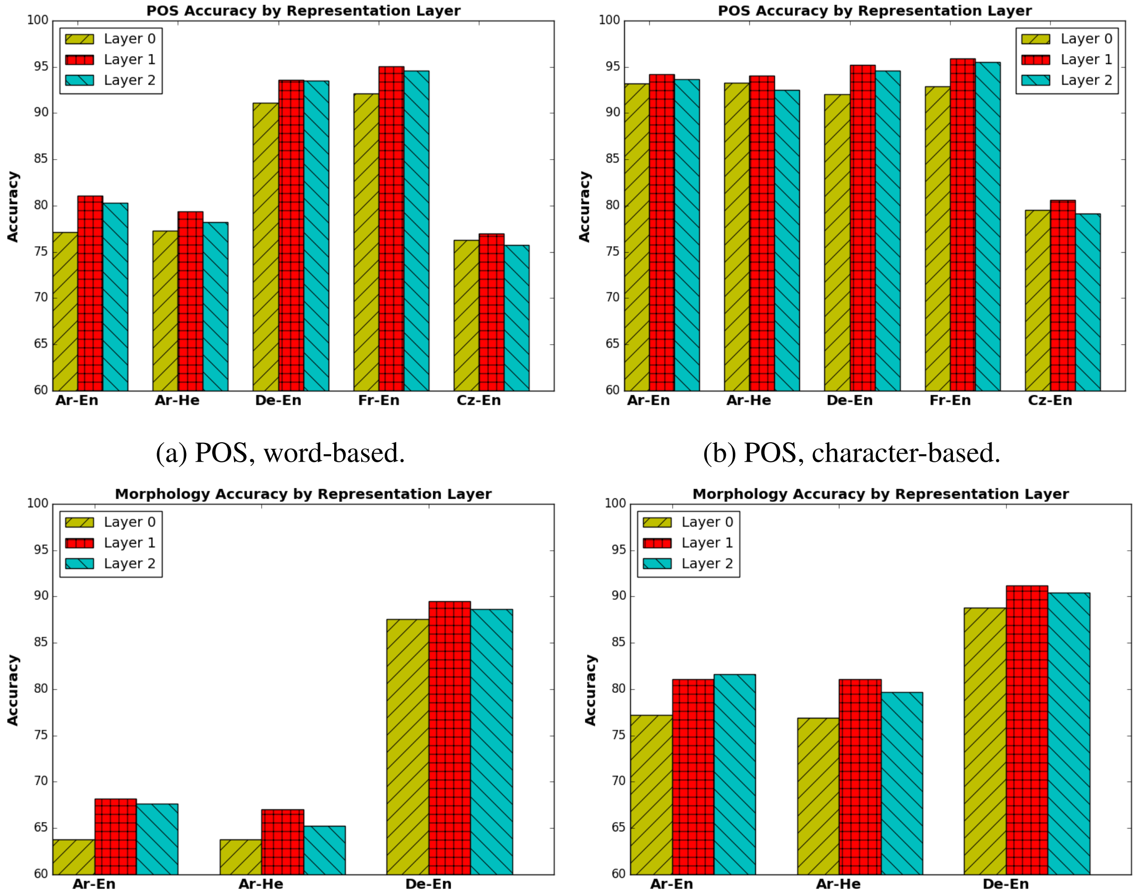
Figure 2-3: Confusion matrices for POS tagging using word-based **(a)** and character-based representations **(b)**.

[10] Arabic examples use the Buckwalter transliteration [48, 140]: `http://www.qamus.org/transliteration.htm`.

2.5.2 Effect of encoder depth

Modern NMT systems use very deep architectures with up to 8 or 16 layers [344, 361]. In order to understand what kind of information different layers capture, different classifiers can be trained on the representations $\boldsymbol{h}_t^{S,l}$ from different layers. The experiments here focus on the case of a 2-layer encoder-decoder model for simplicity, that is, $l \in \{0, 1, 2\}$, where $l = 0$ is the word embedding layer (the input to the encoder).

Figure 2-4 shows POS and morphological tagging results using representations from different encoding layers across five language pairs. The general trend is that passing word vectors through the encoder improves POS and morphological tagging, which can be explained by the contextual information contained in the representations after one layer. However, it turns out that representations from the first layer are better than those from the second layer, at least for the purpose of capturing word structure. In contrast, BLEU scores actually increase when training 2-layer vs. 1-layer models (e.g., +1.11/+0.56 BLEU for Arabic-Hebrew word/character-based models). Thus translation quality improves when adding layers but morphology quality degrades. Intuitively, it seems that lower layers of the network learn to represent word structure while higher layers are more focused on word meaning. This hypothesis will be revisited in Chapter 3. For now, note that a similar pattern was observed in a joint language-vision deep recurrent network [123].



(a) POS, word-based.

(b) POS, character-based.

(c) Morphology, word-based.

(d) Morphology, character-based.

Figure 2-4: The effect of layer depth on POS and morphological tagging using representations from word-based and character-based encoders of different language pairs. Layer 1 tends to perform better than layer 0 (word or character CNN representations) or layer 2.

2.5.3 Effect of target language

While translating from morphologically-rich languages is a challenging task, translating into such languages is even harder.[11] For instance, the Arabic/Czech to English systems obtain BLEU scores of 24.69/23.2 respectively (Table 2.2), while comparable systems translating English to Arabic/Czech obtain only 13.37/13.9 BLEU. How does the target language affect the learned source language representations? Does translating into a morphologically-rich language require more knowledge about source language morphology? In order to investigate these questions, consider the following experiment. Given a certain source language, train neural machine translation models using different target languages. To make a fair comparison, the models are trained on the intersection of the training data based on the source language. In this way the experimental setup is completely identical: the models are trained on the same Arabic sentences with different translations.

Figure 2-5 shows the result of such an experiment with an Arabic source, and multiple target languages. These target languages represent a morphologically-poor language (English), a morphologically-rich language with similar morphology to the source language (Hebrew), and a morphologically-rich language with different morphology (German). As expected, translating into English is easier than translating into the morphologically-richer Hebrew and German, resulting in higher BLEU scores. Despite their similar morphological systems, translating Arabic to Hebrew is worse than Arabic to German, which can be attributed to the richer Hebrew morphology compared to German. POS and morphology accuracies share an intriguing pattern: the representations that are learned when translating into English are better for predicting POS or morphology than those learned when translating into German, which are in turn better than those learned when translating into Hebrew.

[11] Therefore, machine translation from a morphologically-poor language such as English into a morphologically-rich language typically produces much worse results than in the other direction. See for example the recent WMT evaluation results [41]. More references on translating into morphologically-rich languages are given in [79].

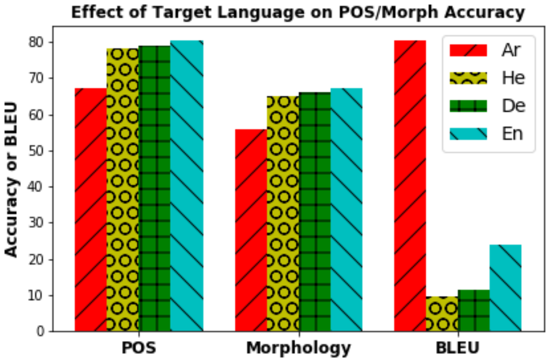


Figure 2-5: Effect of target language on source-side representations in the encoder. POS/morphology accuracy and BLEU scores with Arabic source and different target languages. Translating into a morphologically-poor language leads to slightly improved representations on the source-side.

This is remarkable given that English is a morphologically-poor language that does not display many of the morphological properties that are found in the Arabic source. In contrast, German and Hebrew have richer morphologies, so one could expect that translating into them would make the model learn more about morphology.

A possible explanation for this phenomenon is that the Arabic-English model is simply better than the Arabic-Hebrew and Arabic-German models, as hinted by the BLEU scores in Table 2.2. The inherent difficulty in translating Arabic to Hebrew/German may affect the ability to learn good representations of word structure. However, it turns out that an Arabic-Arabic autoencoder learns to recreate the test sentences extremely well, even though its word representations are actually inferior for the purpose of POS/morphological tagging (Figure 2-5). This implies that higher BLEU does not necessarily entail better morphological representations. In other words, a better translation model learns more informative representations, but only when it is actually learning to translate rather than merely memorizing the data as in the autoencoder case. Note that these trends are consistent in other language pairs (see Table A.1 in Appendix A).

2.6 Decoder Analysis

Languages differ essentially in what they must *convey and not in what they* may *convey.*
— Roman Jakobson, On Linguistic Aspects of Translation

So far we only looked at the encoder. However, the decoder is a crucial part in a neural machine translation system with access to both source and target sentences. Intuitively, the decoder needs to generate grammatical surface forms in the target language, so we may expect it to learn good morphological representations on the target language. This section examines what the decoder learns about morphology of the target language, by following the same methodology. First, a neural machine translation system is trained on the parallel corpus. Then, the trained model is used to encode a source sentence and generate feature representations for words in the target sentence: $\boldsymbol{h}_t^T$, in the notation from Section 1.5.2.[12] These features are used to train a classifier on POS or morphological tagging on the target side.[13] See Figure 2-6 for an illustration of this approach.

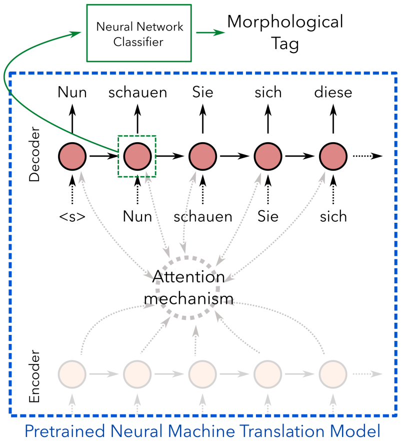


Figure 2-6: Illustration of the approach for analyzing decoder representations. A classifier is trained to predict morphological tags on the target side using features from the decoder of a pre-trained neural machine translation model.

[12] Note that in this case the decoder is given the correct target words one-by-one, similar to the usual neural machine translation training regime.

[13] This section only considers predicted tags for lack of available parallel data with gold POS or morphological tags.

Table 2.3a (1st row) shows the results of using word representations generated with the encoder and the decoder from the Arabic-English and English-Arabic models, respectively. There is a modest drop in representation quality with the decoder. This drop may be correlated with lower BLEU scores when translating English to Arabic vs. Arabic to English. We observed fairly small drops with higher quality translation directions (compare Table 2.3b with Table 2.2).

		POS Accuracy		BLEU	
	Attention	Encoder	Decoder	Arabic-English	English-Arabic
Word	✓	89.62	86.71	24.69	13.37
	✗	74.10	85.54	11.88	5.04
Char	✓	95.35	91.11	28.42	13.00

(a) Arabic POS tagging accuracy using encoder and decoder representations from Arabic-English and English-Arabic models, respectively.

	English-German	English-Czech	German-English	French-English
POS	94.29	71.87	93.26	94.36
BLEU	23.4	13.9	29.6	37.8

(b) POS accuracy and BLEU using (word-based) decoder representations in different language pairs.

Table 2.3: Decoder vs. encoder representations.

The little gap between encoder and decoder representations may sound surprising, when we consider the fundamental tasks of the two modules. The encoder's task is to create a generic, close to language-independent representation of the source sentence, as shown by recent evidence from multilingual NMT [165]. The decoder's task is to use this representation to generate the target sentence in a specific language. One might conjecture that it would be sufficient for the decoder to learn a strong language model in order to produce morphologically-correct output, without learning much about morphology, while the encoder needs to learn quite a lot about source language morphology in order to create a good generic representation. However, their performance seems more or less comparable.[14] The next section investigates what the role of the attention mechanism in the division of labor between encoder and decoder.

2.6.1 Effect of attention

Consider the role of the attention mechanism in learning useful representations: during decoding, the attention weights are combined with the decoder's hidden states to generate the current translation. These two sources of information need to jointly point to the most relevant source word(s) and predict the next most likely word. Thus, the decoder puts significant emphasis on mapping back to the source sentence, which may come at the expense of obtaining a meaningful representation of the current word. A plausible hypothesis, then, is that the attention mechanism hurts the quality of the target word representations learned by the decoder.

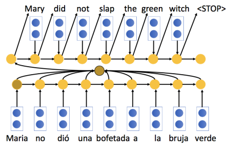


Illustration of attention weights when predicting "Mary"

To test this hypothesis, we train NMT models with and without attention and compare the quality of their learned representations. As TTable 2.3a shows (compare 1st and 2nd rows), removing the attention mechanism decreases the quality of the encoder representations significantly, but only mildly hurts the quality of the decoder representations. It seems that the decoder does not rely on the attention mechanism to obtain good target word representations, contrary to our hypothesis. To evaluate the role of the attention directly, recall that the attention mechanism forms a link between the encoder and decoder that enables the decoder to utilize information from the encoder. Indeed, one can track the attention weights and find the most-attended word during decoding. Adding the encoder representation of this most-attended word to the decoder representation improves the target-side morphological prediction (Figure 2-7), showing that this information can

[14] We will return to this question in Section 2.7, where we attempt to improve the decoder by injecting morphological information while training the NMT system.

be utilized by the decoder through the attention mechanism.

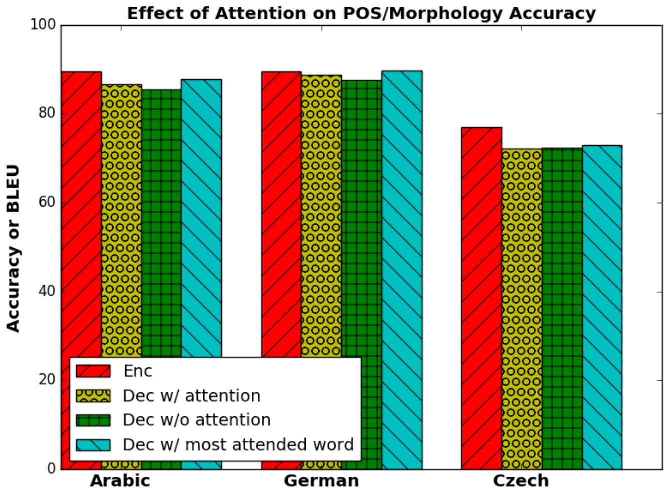


Figure 2-7: Effect of attention mechanism on decoder representations in Arabic POS tagging, German morphological tagging, and Czech morphological tagging. Removing the attention mechanism leads to little or no effect on decoder representations. Including the encoder representation of the most attended to word results in better representations.

2.6.2 Effect of word representation

In the encoder analysis (Section 2.5), character-based representations proved to be better than word-based ones, both in terms of morphology and in overall translation quality. Does this behavior arise also in the decoder? Table 2.3a shows POS accuracy of word-based vs. character-based representations in the encoder and decoder (compare 1st and 3rd rows). In both bases, char-based representations perform better.[15] BLEU scores behave differently: the character-based model leads to better translations in Arabic-to-English, but not in English-to-Arabic. A possible explanation for this phenomenon is that the decoder's

[15] Note that character-based representations in the decoder are applied only on input words. The decoder predictions are still done at the word level, so it is possible to use its hidden states as word representations. Fully-character models [194, 349] go beyond that, but analyzing their representations is less straightforward.

predictions are still done at word level even with the character-based model (which encodes the target input but not the output). In practice, this can lead to generating unknown words. Indeed, in the Arabic-to-English case, the character-based model reduces the number of generated unknown words in the test set by 25%, while in the English-to-Arabic case the number of unknown words remains roughly the same between word-based and character-based models.

2.7 Closing the Loop: Improving the NMT Decoder

Section 2.6 demonstrated that the decoder learns morphological representations of similar or slightly lower quality to that of the decoder. We have also seen that the decoder can utilize information from the encoder through the attention mechanism, while the encoder suffers more from the lack of attention. However, it is not clear whether the decoder is learning just enough morphology or whether translation performance can benefit from improving morphological learning in the decoder. Therefore, this section studies the following question: can the translation performance be improved by injecting morphological information to the neural machine translation decoder?

2.7.1 Methods

Three different methods for promoting morphological awareness in the decoder were investigated (see Figure 2-8). First, a simple *joint generation* approach concatenates the target words and morphological tags. Given a source sentence, the decoder first predicts the target words and then continues to predict the target tags. Second, the example sentences and tag sequences are mixed in the corpus in a *joint learning* approach. In this method, a source sentence is prefixed with a special symbol indicating whether it is to be

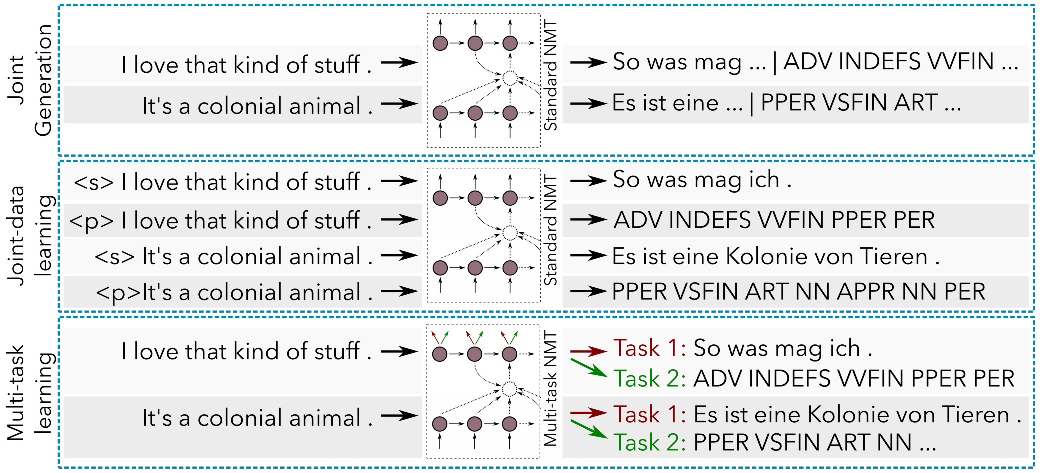

Joint Generation
I love that kind of stuff .
It's a colonial animal .
Standard NMT
So was mag ... | ADV INDEFS VVFIN ...
Es ist eine ... | PPER VSFIN ART ...
Joint-data learning
<s> I love that kind of stuff .
<p> I love that kind of stuff .
<s> It's a colonial animal .
<p>It's a colonial animal .
Standard NMT
So was mag ich .
ADV INDEFS VVFIN PPER PER
Es ist eine Kolonie von Tieren .
PPER VSFIN ART NN APPR NN PER
Multi-task learning
I love that kind of stuff .
It's a colonial animal .
Multi-task NMT
Task 1: So was mag ich .
Task 2: ADV INDEFS VVFIN PPER PER
Task 1: Es ist eine Kolonie von Tieren .
Task 2: PPER VSFIN ART NN ...

Figure 2-8: Methods for injecting morphological knowledge into the decoder: *joint generation* of a sequence of words and morphological tags; *joint data* learning on a corpus with both translations and target-side tags; and *multi-task* learning of translation and morphological tagging.

translated into target words or tags [165, 303]. Third, in a *multi-task learning* approach, the model is modified such that the decoder has two different output layers, one for generating target words and one for generating target tags. The lower parts of the decoder (i.e., the recurrent neural network (RNN) layers) are shared between the tasks, as are the encoder and attention modules. This method optimizes a joint loss function:

$$(1-\lambda)\sum_i \sum_{j=1}^{|t^{(i)}|} \log P(u_j^{(i)}|u_1^{(i)},\ldots,u_{j-1}^{(i)},s^{(i)}) + \lambda \sum_i \sum_{j=1}^{|t^{(i)}|} \log P(m_j^{(i)}|m_1^{(i)},\ldots,m_{j-1}^{(i)},s^{(i)}) \tag{2.1}$$

where $m_j^{(i)}$ is the j-th tag in the i-th target sentence. As before, $s^{(i)}$ is the i-th source sentence, $t^{(i)}$ is the i-th target sentence, and $u_j^{(i)}$ is the j-th word in the i-th target sentence. Here λ is a hyper-parameter that provides a trade-off between morphology (higher values) and translation (lower values).

2.7.2 Experiments

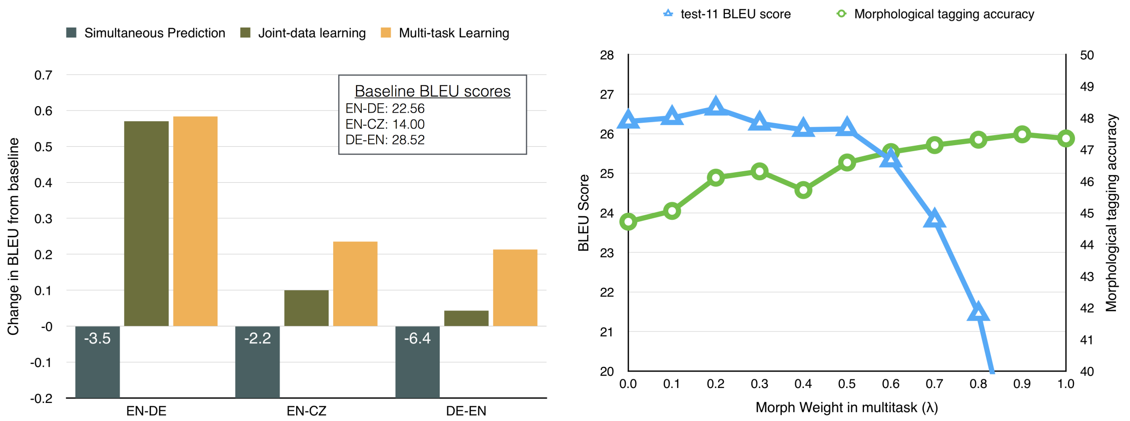
The different methods for improving morphological learning were tested on two language pairs where the target language has rich morphology (English-German and English-Czech) and one pair where the target language is morphologically-poor (German-English). The experimental setup follows that described in 2.4, using the same datasets and baseline neural machine translation systems.[16]

Figure 2-9a shows the improvement in BLEU when adding morphology to the decoder using the three methods, compared to the baseline systems. Clearly, the joint-generation is unsuccessful in improving translation performance. This may be because concatenating target words and tags leads to large distances between each word and its corresponding tag.[17] The joint-learning approach is more successful, leading to +0.6 on English-German, but little to no improvements on the other language pairs. The multi-task learning approach seems the best of the three. It too obtains +0.6 on English-German, but also slightly improves the other language pairs by about +0.2 BLEU.

The multi-task learning results in Figure 2-9a are the test results corresponding to the best value of λ, as tuned on the held-out tune set. Figure 2-9b shows an example of such tuning for English-German, demonstrating the trade-off between morphology and translation prediction. In all language pairs, a value of $\lambda = 0.2$ produced the best translation performance on the tune set, indicating that a modest amount morphological knowledge is helpful for translation.

[16] The only difference is that test-11 is used for tuning, while the other test sets are used for evaluation.

[17] One remedy may be to interleave words and tags [250].



(a) Improvements from adding morphology. A y-value of zero represents the baseline.

(b) Multi-task learning: translation vs. morphological tagging weight for the En-De model.

Figure 2-9: Effect of adding morphology to the decoder in different ways.

2.7.3 Discussion

The experiments reported in this section provide a good example of how analysis work can lead to insights that improve the original end-to-end system. The analysis in Sections 2.5 and 2.6 revealed that both the neural machine translation encoder and the decoder learn quite informative representations in terms of morphology, while the attention mechanism is important for encoder representation quality more than for decoder representation quality. This discovery motivated the investigation of several different methods for improving neural machine translation by injecting morphological knowledge to the decoder. This is therefore a fine example of closing the loop that was introduced in Section 1.2 (recall Figure 1-2), connecting analysis back into architecture changes in the original system.

2.8 Conclusion and Future Work

The representations used by neural networks for linguistic units are crucial for obtaining high-quality translation. This chapter investigated how neural machine translation models learn word structure. Their representation quality was evaluated on POS and morphological tagging in a number of languages. The results lead to the following conclusions:

- Character-based representations are better than word-based ones for learning morphology, especially in rare and unseen words.

- Lower layers of the neural network are better at capturing morphology, while deeper networks improve translation performance. This led to the hypothesis that lower layers are more focused on word structure, while higher ones are focused on word meaning. This idea will be explored in the next chapter.

- The target language impacts how well the encoder learns source language morphology. Translating into morphologically-poorer languages leads to better source-side word representations. This is partly, but not completely, correlated with BLEU scores.

- There are only little differences between encoder and decoder representation quality. The attention mechanism does not seem to significantly affect the quality of the decoder representations, while it is important for the encoder representations. These results motivated jointly learning translation and morphology, which led to improved representations and translation quality.

The next chapter will revisit some of these questions in the context of a lexical semantic task, with a particular focus on questions of representation depth.

These insights can guide further development of neural MT systems. For instance, future work can investigate the incorporation of morphology into other parts of the neural machine translation architecture. Jointly learning translation and morphology is a promising direction. The analysis in this chapter indicates that this kind of approach should take into account factors such as the encoding layer and the type of word representation. Another area for future work is to extend the analysis to other word representations (such as byte-pair encoding or the word-piece model), deeper networks, and to study other languages that exhibit rich morphological systems. Finally, a similar methodology can be applied for studying morphological properties in other “end-to-end” neural network models, such as syntactic parsing, coreference resolution, and more high-level language understanding tasks.

Chapter 3

Word Meaning and Neural Machine Translation: Lexical Semantics

“every word (lexical unit) has also something that is individual, that makes it different from any other word. And it is just the lexical meaning which is the most outstanding individual property of the word.”
— Ladislav Zgusta, Manual of Lexicography

3.1 Introduction

A core ingredient of the translation process is capturing the meaning of individual words – that is, *lexical semantics* – and rendering them in a target language. In most approaches to machine translation, such meaning is acquired automatically from a parallel corpus of source and target sentences, without providing direct supervision of word meaning. However, some studies incorporate lexical semantic information in machine translation systems, for instance by using word sense disambiguation [52, 58, 341]. Although recent studies on neural machine translation incorporate such information either explicitly [285] or implicitly [215], most neural machine translation systems do not utilize semantic information, instead relying on the model acquiring the necessary meaning representations from the parallel corpus.

Super-word levels
texts
utterances
sentences
phrases
relations
words
morphemes
syllables
characters
phonemes
Sub-word levels

This chapter studies how information on word meaning in captured in neural machine translation in the context of a lexical semantic (SEM) tagging task, introduced in [39] . It is a sequence labeling task: given a sentence, the goal is to assign to each word a tag representing a semantic class. This is a good task to use as a starting point for investigating semantics because: *i)* tagging words with semantic labels is very simple, compared to building complex relational semantic structures; *ii)* it provides a large supervised dataset to train on, in contrast to most of the available datasets on word sense disambiguation, lexical substitution, and lexical similarity; and *iii)* the proposed SEM tagging task is an abstraction over part-of-speech (POS) tagging aimed at being language-neutral, and oriented to multilingual semantic parsing, all relevant aspects to machine translation. The following is a brief overview of the task and its associated dataset; refer to [1, 39] for more details.

The semantic classes abstract over redundant POS distinctions and disambiguate useful cases inside a given POS tag. For instance, proximal and distal demonstratives (e.g., *this* and *that*) are typically assigned the same POS tag (DT) but receive different SEM tags (PRX and DST, respectively), and proper nouns are disambiguated into several classes such as geo-political entity, location, organization, person, and artifact. Other examples of SEM tag distinctions include determiners like *every*, *no*, and *some* that are typically assigned a single POS tag (e.g., DT in the Penn Treebank), but have different SEM tags, reflecting universal quantification (AND), negation (NOT), and existential quantification (DIS), respectively. The comma, whose POS tag is a punctuation mark, is assigned different SEM tags representing conjunction, disjunction, or apposition, according to its discourse function. Other nouns are divided into “role” entities (e.g., *boxer*) and “concepts” (e.g., *wheel*), a distinction reflecting existential consistency: an entity can have multiple roles but cannot be two different concepts.

As a motivating example, consider pronouns like *myself*, *yourself*, and *herself*. They may have reflexive or emphasizing functions, as in (1a) and (2a), respectively. In these examples, *herself* has the same POS tag (`PRP`) but different SEM tags: `REF` for a reflexive function and `EMP` for an emphasizing function.

(1) a. Sarah bought *herself* a book

b. Sarah *se* compró un libro

(2) a. Sarah *herself* bought a book

b. Sarah *misma* compró un libro

Capturing semantic distinctions of this sort can be important for producing accurate translations. For instance, example (1a) would be translated into Spanish with the reflexive pronoun *se* (example 1b), whereas example (2a) would be translated with the intensifier *misma* (example 2b). Therefore, a machine translation system needs to learn different representations of *herself* in the two sentences.

This chapter studies how this sort of semantic information is captured in the neural machine translation system by answering the following specific questions:

1. Do neural machine translation systems learn informative semantic representations?

2. What parts of the system learn more about SEM tagging? Chapter 2 found that POS and morphology information is better captured at lower layers of the neural machine translation encoder. Is the same true for SEM tagging information?

3. What is the effect of the target language when learning source-side representations for these tasks? Is SEM tagging more or less affected by the target language compared to morphological tagging (Chapter 2)?

To answer these questions, I exploit the semantic tagging task described above. I generate representations from a variety of neural machine translation models, and train classifiers to predict semantic tags. I compare the performance of representations generated by the same translation models on a POS tagging task. The analysis yields the following insights regarding representation learning in neural machine translation:

- Consistent with the results from Chapter 2, I find that lower layer representations are usually better for POS tagging. However, I also find that representations from higher layers of the neural machine translation encoder are better at capturing lexical semantics, even though these are word-level labels. This is especially true with tags that are more semantic in nature such as discourse functions and noun concepts. An error analysis shows how predicting such tags require more contextual information.

- I also observe little effect of the target language on source-side representation, in contrast to the results on morphology from Chapter 2. A more careful investigation reveals that the effect of target language diminishes as the size of data used to train the neural machine translation model increases.

3.2 Related Work

Prior work has considered integrating lexical semantic information in machine translation systems by using word sense disambiguation [52, 58, 341], and recent work integrated sense embeddings [285] or other methods for improving sense disambiguation in neural machine translation [215]. However, as statistical machine translation systems are contextual by design, it is thought that they do not typically require special word disambiguation treatment [181].

A variety of other semantic properties have been considered in the machine translation literature, most prominently semantic roles [119, 214, 341, inter alia] and predicate-argument structure [189, 205, 343, 346]. These, however, operate above the word level. Chapter 4 explores such properties in neural machine translation.

On the analysis side, recent work has considered how word senses are captured in neural machine translation by evaluating systems on contrastive pairs [285], or by visualizing representations and measuring their disambiguation quality [228]. Hill et al. [148] analyzed word embeddings in neural machine translation models and found that they outperform monolingual word embeddings on semantic similarity tasks. They also observed a limited effect of the target language on source-side word embeddings.

3.3 Methodology

The methodological approach used in this chapter follows the high-level approach presented in Section 1.2, adapted for SEM tagging. Recall the following three steps: *(i)* train a neural machine translation system on a parallel corpus; *(ii)* generate feature representations for words using the trained model; and *(iii)* train a classifier using the generated features to make predictions for a SEM tagging task. The classifier accuracy on the test set is used for evaluating the quality of the neural machine translation representations. In order to compare semantic (SEM) and part-of-speech (POS) information, a separate classifier is trained on POS tagging. Figure 3-1 illustrates this process.

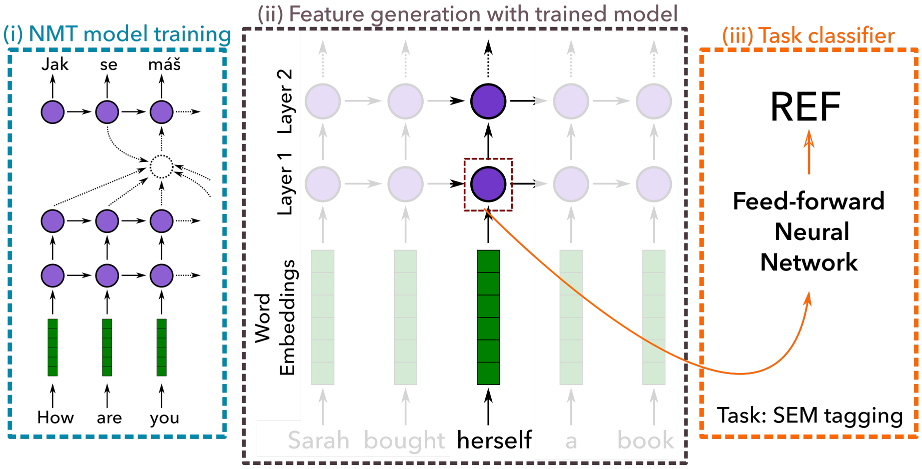


Figure 3-1: Illustration of the approach for studying SEM and POS tagging: *(i)* a neural machine translation system trained on parallel data; *(ii)* features are generated with the pre-trained model; *(iii)* a classifier is trained using the generated features. Here a classifiers is trained on SEM tagging using features from the first encoding layer.

The neural machine translation architecture is a recurrent neural network (RNN) encoder-decoder model with attention, as described in detail in Section 2.3, with the following differences. First, the majority of the experiments in this section are conducted with a deeper, 4-layer model. This is made possible by training the neural machine translation systems on a larger parallel corpus. Additional experiments compare the results to shallower models. Second, three different encoders are considered: unidirectional, bidirectional, and an encoder with residual connections [145, 344].

The classifier is exactly the same as used in Chapter 2, that is, a one-hidden layer neural network whose input is the encoder representation at a particular layer, $\boldsymbol{h}_i^{S,l}$, and whose output is the label set. See Section 2.3 for more details. Note that this chapter only studies encoder-side representations.[1]

[1] Investigating the representations on the decoder side would require having either good automatic taggers or a parallel corpus with annotation on the target side. Progress in developing tools [39] and resources [1] for SEM tagging may prove useful in the future.

3.4 Data

The experiments reported in this section on SEM and POS tagging are all conducted on English, as the SEM tagging task and dataset are recent developments that were initially only available in English [39].[2]

MT data Neural machine translation systems are trained on the fully-aligned United Nations corpus [362], which includes 11 million multi-parallel sentences in six languages: Arabic (Ar), Chinese (Zh), English (En), French (Fr), Spanish (Es), and Russian (Ru). The experiments are conducted with English-to-* models, trained on the first 2 million sentences of the training set, and using the official train/dev/test split. This dataset has the benefit of multiple alignment of the six languages, which allows for comparable cross-linguistic analysis. Note that the parallel dataset is only used for training the neural machine translation model. The classifier is then trained on the supervised data (described next) and all accuracies are reported on the English test sets.

The texts are preprocssed with the tokenization script provided with the Moses machine translation toolkit [186]. The Chinese dataset is segmented with the Stanford word segmenter [61, 323].

Annotated data The SEM tagging dataset includes 66 fine-grained tags grouped in 13 coarse categories. The experiments are conducted on the silver part of the dataset. See Table 3.1a for representative statistics, and refer to [1, 39] for more details.

The POS tagging dataset is based on the Penn Treebank [225] with the standard split: parts 2–21/22/23 for train/dev/test. See Table 3.1a for statistics. There are 34 POS tags.

[2] Subsequent to this work, the Groningen Parallel Meaning Bank (PMB) [1] has added annotations in German, Dutch, and Italian: `http://pmb.let.rug.nl`. It thus opens possibilities for future work comparing representations in multiple languages from the perspective of SEM tagging.

		Train	Dev	Test
POS	Sentences	38K	1.7K	2.3K
	Tokens	908K	40K	54K
SEM	Sentences	42.5K	6.1K	12.2K
	Tokens	937K	132K	266K

(a) Dataset statistics.

	MFT	UnsupEmb	Word2Tag
POS	91.95	87.06	95.55
SEM	82.00	81.11	91.41

(b) Baselines and an upper bound.

Ar	Es	Fr	Ru	Zh	En
32.7	49.1	38.5	34.2	32.1	96.6

(c) BLEU scores.

Table 3.1: **(a)** Statistics of the part-of-speech (POS) and semantic (SEM) tagging datasets. **(b)** Tagging accuracy with the most frequent tag baseline (MFT), a classifier using unsupervised word embeddings (UnsupEmb), and an upper bound encoder-decoder (Word2Tag). **(c)** BLEU scores for machine translation systems trained on an English source and different target languages: Arabic (Ar), Spanish (Es), French (Fr), Russian (Ru), Chinese (Zh), and an English autoencoder (En).

3.4.1 Baselines and an upper bound

Table 3.1b shows the results of two baselines: assigning to each word the most frequent tag (MFT) according to the training data (with the global majority tag for unseen words); and training with unsupervised word embeddings (UnsupEmb) as features for the classifier, which shows what a simple task-independent distributed representation can achieve.[3] The UnsupEmb baseline performs rather poorly on both POS and SEM tagging, even below the most frequent tag baseline (MFT), indicating that non-contextual, unsupervised word embeddings are poor representations for POS and SEM tags. The table also reports an upper bound of training an encoder-decoder on word-tag sequences (Word2Tag), simulating what an NMT-style model can achieve by directly optimizing for the tagging tasks.

[3] The unsupervised word embeddings were trained with a Skip-gram negative sampling model [238] with 500 dimensional vectors on the English side of the parallel data, to mirror the NMT word embedding size.

	POS Tagging Accuracy					SEM Tagging Accuracy				
	0	1	2	3	4	0	1	2	3	4
Arabic	88.0^{*}	92.4	91.9^{*}	92.0^{*}	92.1^{*}	81.9^{*}	87.9	87.4^{*}	87.8	88.3^{*}
Spanish	87.9^{*}	91.9	91.8	92.3^{*}	92.4^{*}	81.9^{*}	87.7	87.5^{*}	87.9^{*}	88.6^{*}
French	87.9^{*}	92.1	91.8	92.1	92.5^{*}	81.8^{*}	87.8	87.4^{*}	87.9^{**}	88.4^{*}
Russian	87.8^{*}	92.1	91.8^{*}	91.6^{**}	92.0	81.8^{*}	87.9	87.3^{*}	87.3^{*}	88.1^{*}
Chinese	87.7^{*}	91.5	91.3	91.2^{*}	90.5^{*}	81.8^{*}	87.7	87.2^{*}	87.3^{*}	87.7^{*}
English	87.4^{*}	89.4	88.3	87.9^{*}	86.9^{*}	81.2^{*}	84.5	83.2^{*}	82.9^{*}	82.1^{*}

Table 3.2: SEM and POS tagging accuracy on English using features generated by different encoding layers of 4-layered neural machine translation models trained with different target languages. "English" row is an autoencoder. Statistically significant differences from layer 1 are shown at $p < 0.001^{(*)}$ and $p < 0.01^{(**)}$.

3.5 Effect of Depth

Recall the results in Section 2.5 regarding the effect of depth on representation quality: lower layers of the neural machine translation encoder generated better representations for POS and morphological tagging. This section investigates the quality of representations at different encoding layers, from the perspective of SEM tagging. The results are also compared to POS tagging. The primary research question is whether a higher-level task like SEM tagging would be better represented at higher layers.

Table 3.2 summarizes the results of training classifiers to predict POS and SEM tags using features generated by different encoding layers of 4-layered neural machine translation systems. In the POS tagging results (first block), as the representations move above layer 0, performance jumps to around 91–92%. This is above the UnsupEmb baseline but only on par with the MFT baseline (Table 3.1b). The results are also far below the Word2Tag upper bound (Table 3.1b).

Comparing layers 1 through 4, in 3/5 target languages (Arabic, Russian, and Chinese), POS tagging accuracy peaks at layer 1 and does not improve at higher layers, with some drops at layers 2 and 3. In 2/5 cases (Spanish, French) the performance is higher at layer 4. This result is partially consistent with the results from Section 2.5 and with previous findings regarding the quality of lower layer representations for the POS tagging task [306]. One possible explanation for the discrepancy when using different target languages is that French and Spanish are typologically closer to English compared to the other languages. It is possible that when the source and target languages are more similar, they share similar POS characteristics, leading to more benefit in using upper layers for POS tagging.

Turning to SEM tagging (Table 3.2, second block), representations from layers 1 through 4 boost the performance to around 87–88%, far above the UnsupEmb and MFT baselines. While these results are below the Word2Tag upper bound (Table 3.1b), they indicate that neural machine translation representations contain useful information for SEM tagging.

Going beyond the 1st encoding layer, representations from layers 2 and 3 do not consistently improve semantic tagging performance. However, representations from the layer 4 lead to significant improvement with all target languages except for Chinese. Note that there is a statistically significant difference ($p < 0.001$) between layers 0 and 1 for all target languages, and between layers 1 and 4 for all languages except for Chinese, according to the approximate randomization test [265].

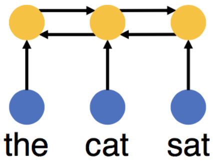
Intuitively, higher layers have a more global perspective because they have access to higher representations of the word and its context, while lower layers have a more local perspective. Layer 1 has access to context but only through one hidden layer which may not be sufficient for capturing semantics. It appears that higher representations are necessary for learning even relatively simple lexical semantics.

Finally, the results show that English-English encoder-decoders (that is, English autoencoders) produce poor representations for POS and SEM tagging (last row in Table 3.2). This is especially true with higher layer representations (e.g., around 5% below the machine translation models using representations from layer 4). In contrast, the autoencoder has excellent sentence recreation capabilities (96.6 BLEU, Table 3.1c). This indicates that learning to translate (to any foreign language) is important for obtaining useful representations for both tagging tasks. These results are consistent with the findings reported in Section 2.5 regarding morphology.

3.5.1 Other architectural variants

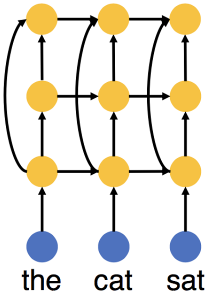
The results reported in Table 3.2 are with a unidirectional encoder. In order to confirm that the observed patterns hold in different architectures, the following experiments consider two architectural variants that have been shown to benefit neural machine translation systems, bidirectional encoder and residual connections, as well as systems trained with different depths.

Bidirectional long short-term memorys (LSTMs) have become ubiquitous in natural language processing (NLP) and also give some improvement as neural machine translation encoders [43]. The experiments conducted here confirm these results and produce improvements in both translation (+1–2 BLEU) and SEM tagging quality (+3–4% accuracy), across the board, when using a bidirectional encoder. Some of the bidirectional models obtain 92–93% accuracy, which is close to the state-of-the-art on this task [39]. Similar improvements were observed on POS tagging. Comparing POS and SEM tagging (Table 3.3a) shows that higher layer representations improve SEM tagging, while POS tagging peaks at layer 1, in line with the findings with a unidirectional encoder.



Bidirectional RNN

Deep networks can sometimes be trained better if residual connections are introduced between layers. Such connections were also found useful for SEM tagging [39]. Indeed, residual connections lead to small but consistent improvements in both translation (+0.9 BLEU) and POS and SEM tagging (up to +0.6% accuracy) (Table 3.3a). Similar trends



Residual RNN

arise as before: POS tagging does not benefit from features from the upper layers, while SEM tagging improves with layer 4 representations.

In comparing network depth in NMT, encoders with 2 to 4 layers tend to perform the best [43]. Table 3.3b shows consistent trends using models trained originally with 2, 3, and 4 layers: POS tagging does not benefit from upper layers, while SEM tagging does, although the improvement is rather small in the shallower models.

		0	1	2	3	4
Uni	POS	87.9	**92.0**	91.7	91.8	91.9
	SEM	81.8	87.8	87.4	87.6	**88.2**
Bi	POS	87.9	**93.3**	92.9	93.2	92.8
	SEM	81.9	91.3	90.8	**91.9**	**91.9**
Res	POS	87.9	**92.5**	91.9	92.0	92.4
	SEM	81.9	88.2	87.5	87.6	**88.5**

(a) Comparing representations from different layers of unidirectional, bidirectional, and residual encoders.

		0	1	2	3	4
4	POS	87.9	**92.0**	91.7	91.8	91.9
	SEM	81.8	87.8	87.4	87.6	**88.2**
3	POS	87.9	**92.5**	92.3	92.4	–
	SEM	81.9	88.2	88.0	**88.4**	–
2	POS	87.9	**92.7**	**92.7**	–	–
	SEM	82.0	88.5	**88.7**	–	–

(b) Comparing representations from different layers of models originally trained with 2/3/4-layer encoders.

Table 3.3: POS and SEM tagging accuracy using different neural machine translation architectures **(a)** and depths **(b)**. Results are accuracies averaged over all non-English target languages. The best result in each row is shown in **bold**.

3.6 Effect of Target Language

Does translating into different languages make the NMT system learn different source-side representations? Section 2.5 reported a fairly consistent effect of the target language on the quality of encoder representations for POS and morphological tagging, with differences of $\sim$2–3% in accuracy. This section examines if such an effect exists in both POS and SEM tagging.

Table 3.2 also shows results using features obtained by training neural machine translation systems on different target languages (the English source remains fixed). In both POS and SEM tagging, there are very small differences with different target languages (∼0.5%), except for Chinese which leads to slightly worse representations. While the differences are small, they are mostly statistically significant. For example, at layer 4, all the pairwise comparisons with different target languages are statistically significant ($p < 0.001$) in SEM tagging, and all except for two pairwise comparisons (Arabic vs. Russian and Spanish vs. French) are significant in POS tagging.

The effect of the target language is much smaller than that observed in Section 2.5 for POS and morphological tagging. This discrepancy can be attributed to the fact that the machine translation systems in this section are trained on much larger corpora (10x), so it is possible that some of the differences disappear when the translation model is of better quality. To verify this, consider the results in Table 3.4, where the systems were trained using a smaller data size (200K sentences), comparable to the size used in Section 2.5. In the smaller data scenario, there is a variance in classifier accuracy of 1–2%, based on target language, which is consistent with Section 2.5. This is true for both POS and SEM tagging. The differences in POS tagging accuracy are statistically significant ($p < 0.001$) for all pairwise comparisons except for Arabic vs. Russian. The differences in SEM tagging accuracy are significant for all comparisons except for Russian vs. Chinese.

Figure 3-2 shows that these trends hold in different layers. Representations from a model trained on less data (200K sentences) are more sensitive to the target language at all encoder layers, and especially at the very high layers. Larger training data leads to less sensitive representations, but 2 million sentences seem to be sufficient for this. Models trained on much more data (the full 11m sentences dataset) are about as sensitive to the target language as those trained on 2m sentences, which is the main setting used throughout this chapter.

Finally, note that training an English autoencoder on the smaller dataset results in much worse representations compared to machine translation models, for both POS and SEM tagging (Table 3.4, last column), consistent with the behavior on the larger data (Table 3.2, last column).

	POS	SEM
Arabic	88.7	85.3
Spanish	90.0	86.1
French	89.6	85.8
Russian	88.6	85.2
Chinese	87.4	85.0
English	85.2	80.7

Table 3.4: POS and SEM tagging accuracy using features generated from the 4th encoding layer, trained with different target languages on a smaller corpus (200K sentences).

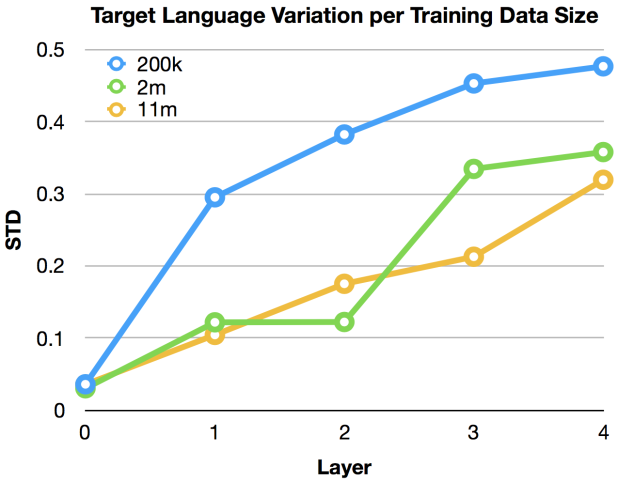


Figure 3-2: Effect of training data size on the variation across target languages when predicting SEM tags. The x-axis shows the layer number. The y-axis shows the standard deviation for all non-English target languages. The representations from a model trained on a small training set (200K sentences) are more sensitive to the target language (larger standard deviations). Higher layer representations exhibit a larger variation across target languages.

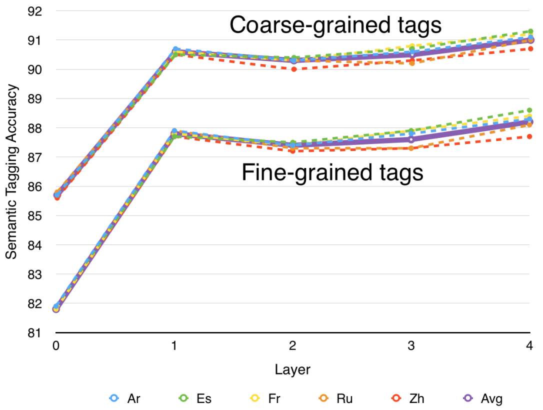


Figure 3-3: SEM tagging accuracy with fine/coarse-grained tags using features generated from different encoding layers of 4-layered neural machine translation models trained with different target languages.

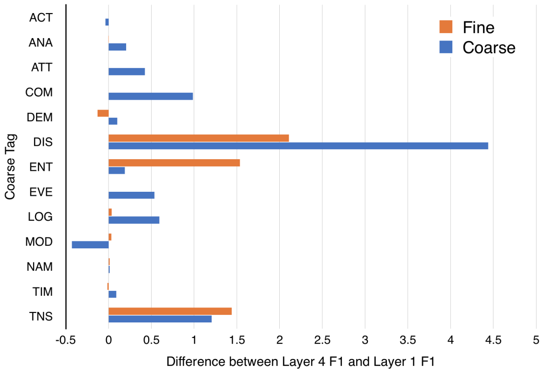


Figure 3-4: Difference in F_1 when using representations from layer 4 compared to layer 1, showing F_1 when directly predicting coarse tags (blue) and when predicting fine-grained tags and averaging inside each coarse tag (red).

3.7 Analysis

3.7.1 Analysis at the semantic tag level

The SEM tags are grouped in coarse-grained categories such as events, names, time, and logical expressions [39]. Figure 3-3 (top lines) shows the results of training and testing classifiers on coarse-grained tags. Similar trends to the fine-grained case arise, with higher absolute scores: significant improvement using the 1st encoding layer and some additional improvement using the 4th layer, both statistically significant ($p < 0.001$). As before, there is a small effect of the target language.

Figure 3-4 shows the change in F_1 score (averaged over target languages) when moving from layer 1 to layer 4 representations. The blue bars describe the differences per coarse tag when directly predicting coarse tags. The red bars show the same differences when predicting fine-grained tags and micro-averaging inside each coarse tag. The former shows the differences between the two layers at distinguishing among coarse tags. The latter gives an idea of the differences when distinguishing between fine-grained tags within a coarse category. The first observation is that in the majority of cases there is an advantage for classifiers trained with layer 4 representations. That is, higher layer representations are better suited for learning the SEM tags, at both coarse and fine-grained levels.

Considering specific tags, higher layers of the model are especially better at capturing semantic information such as *discourse relations* (DIS tag: accounting for subordinate, coordinate, and apposition relations), semantic properties of nouns (*roles* vs. *concepts*, within the ENT tag), *events* and *predicate tense* (EVE and TNS tags), *logic relations* and *quantifiers* (LOG tag: disjunction, conjunction, implication, existential, universal, etc.), and *comparative constructions* (COM tag: equatives, comparatives, and superlatives). These examples represent semantic concepts and relations that require a level of abstraction going beyond the lexeme or word form, and thus might be better represented in higher layers in the deep network.

One negative example that stands out in Figure 3-4 is the prediction of the MOD tag, corresponding to *modality* (necessity, possibility, and negation). It seems that such semantic concepts should be better represented in higher layers following our previous hypothesis. Still, layer 1 is better than layer 4 in this case. One possible explanation is that words tagged as MOD form a closed class category, with only a few and mostly unambiguous words (“no”, “not”, “should”, “must”, “may”, “can”, “might”, etc.). It is enough for the classifier to memorize these words in order to predict this class with high F_1, and this is something that occurs better in lower layers. One final case worth mentioning is the NAM category, which stands for different types of named entities (person, location, organization, artifact, etc.). In principle, this seems a clear case of semantic abstractions suited for higher layers, but the results from layer 4 are not significantly better than those from layer 1. This might be signaling a limitation of the neural machine translation system at learning this type of semantic classes. Another factor might be the fact that many named entities are out-of-vocabulary (OOV) words for the neural machine translation system.

3.7.2 Analyzing discourse relations

As shown in Figure 3-4, the largest improvement when going from layer 1 to layer 4 representations is obtained when predicting discourse relations (DIS category). Intuitively, identifying discourse relations requires a relatively large context so it is expected that higher layers would perform better in this case. It is instructive to analyze specific cases of disagreement between predictions using representations from layer 1 and layer 4. There are three discourse relations in the SEM tags annotation scheme: subordinate (SUB), coordinate (COO), and apposition (APP) relations. For each of these, Figure 3-5 (examples 1–9) shows the first three cases in the test set where layer 4 representations correctly predicted the tag but layer 1 representations were wrong. Examples 1–3 have subordinate conjunctions (*as*, *after*, *because*) connecting a main and an embedded clause, which layer 4 is able to correctly predict. Layer 1 mistakes these as attribute tags (REL, IST) that are usually used for prepositions. In examples 4–5, the coordinate conjunction *and* is used to connect sentences/clauses, which layer 4 correctly tags as COO. Layer 1 wrongly predicts the tag AND, which is used for conjunctions connecting shorter expressions like words (e.g., "murder *and* sabotage" in example 1). Example 6 is probably an annotation error, as *and* connects the phrases "lame gait" and "wrinkled skin" and should be tagged as AND. In this case, layer 1 is actually correct. In examples 7–9, layer 4 correctly identifies the comma as introducing an apposition, while layer 1 predicts NIL, a tag for punctuation marks without semantic content (e.g., end-of-sentence period). As expected, in most of these cases identifying the discourse function requires a fairly large context.

Finally, examples 10–12 show the first three occurrences of AND in the test set, where layer 1 was correct and layer 4 was wrong. Interestingly, two of these (10–11) are clear cases of *and* connecting clauses or sentences, which should have been annotated as `COO`, and the last (12) is a conjunction of two gerunds. The predictions from layer 4 in these cases thus appear justifiable.

	L1	L4	
1	REL	*SUB*	Zimbabwe 's President Robert Mugabe has freed three men who were jailed for murder and sabotage *as* they battled South Africa 's anti-apartheid African National Congress in 1988 .
2	REL	*SUB*	The military says the battle erupted *after* gunmen fired on U.S. troops and Afghan police investigating a reported beating of a villager .
3	IST	*SUB*	Election authorities had previously told Haitian-born Dumarsais Simeus that he was not eligible to run *because* he holds U.S. citizenship .
4	AND	*COO*	Fifty people representing 26 countries took the Oath of Allegiance this week (Thursday) *and* became U.S. citizens in a special ceremony at the Newseum in Washington , D.C.
5	AND	*COO*	But rebel groups said on Sunday they would not sign *and* insisted on changes .
6	AND	*COO*	A Fox asked him , " How can you pretend to prescribe for others , when you are unable to heal your own lame gait *and* wrinkled skin ? "
7	NIL	*APP*	But Syria 's president , Bashar al-Assad , has already rejected the commission 's request to interview him .
8	NIL	*APP*	Hassan Halemi , head of the pathology department at Kabul University where the autopsies were carried out , said hours of testing Saturday confirmed the identities of teachers Jun Fukusho and Shinobu Hasegawa .
9	NIL	*APP*	Mr. Hu made the comments Tuesday during a meeting with Ichiro Ozawa , the leader of Japan 's main opposition party .
10	*AND*	COO	In Washington , D.C. , abortion opponents will march past the U.S. Capitol *and* end outside the Supreme Court .
11	*AND*	COO	Van Schalkwyk said no new coal-fired power stations would be approved unless they use technology that captures *and* stores carbon emissions .
12	*AND*	COO	A MEMBER of the Kansas Legislature meeting a Cake of Soap was passing it by without recognition , but the Cake of Soap insisted on stopping *and* shaking hands .

Figure 3-5: Examples of cases of disagreement between layer 1 (L1) and layer 4 (L4) representations when predicting SEM tags. The correct tag is *italicized* and the relevant word is *underlined*.

3.8 Conclusion and Future Work

In this chapter, I explored what kind of linguistic information neural machine translation models learn at different layers , focusing on lexical semantics. The experimental evaluation led to interesting insights about the hidden representations in neural machine translation models:

- POS tagging information is better captured in lower layers of the neural machine translation encoder, while SEM tagging information is represented better at higher layers. This pattern is consistent in various neural machine translation architectures and models.
- Higher layers are especially helpful for capturing tags that are more semantic in nature, such as discourse functions.
- The target language has a small effect on representation quality on the encoder side. With smaller training data, this effect is more pronounced.

Future work can extend this analysis to other lexical semantic tasks, such as word sense disambiguation or word similarity. New large-scale datasets with sense annotations can serve as a good test bed for using the same methodology [87]. Another important direction is to study similar semantic tasks in other languages. Again, having large datasets is key.[4]

Finally, improving neural machine translation by exploiting semantic datasets is still to be explored. I hope that some of the insights in this chapter would guide better integration of lexical semantic knowledge in neural machine translation.

[4] The PMB [1] is a relevant resource. Its recent release includes semantic tags in multiple languages, although the annotated data are still limited in size. See `http://pmb.let.rug.nl`.

Chapter 4

Sentence Structure and Neural Machine Translation: Word Relations

> *"The sentence is an organized whole, the constituent elements of which are words ... Between the word and its neighbors, the mind perceives connections, the totality of which forms the structure of the sentence."*
> *— Lucien Tesnière, Éléments*[a]

[a] Translation from French by J. Nivre [260]

4.1 Introduction

Chapters 2 and 3 studied neural machine translation representations from the perspective of morphology and lexical semantics. These are chiefly word-level properties, and the analysis was therefore limited to word representations that are learned in neural machine

in translation, grammar often has the effect of a straitjacket, forcing the translator along a certain course which may or may not follow that of the source text as closely as the translator would like it to.
— Mona Baker [20]

translation models. However, modeling structure is an important aspect in machine translation. Before defining the kind of structural information that this chapter is concerned with, some background on structure in machine translation is in order.

Early conceptions of machine translation have considered structure to be an important ingredient, and formulated machine translation as a rule-based structure transfer problem [338, 350]. However, as with other early work on machine translation, this approach proved to be unscalable in practice [338].

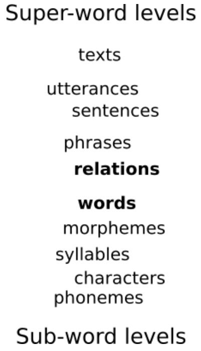
With the rise of statistical machine translation (see Section 1.5.1), many different methods for incorporating syntax have been proposed. An important development is the introduction of phrase-based machine translation (PBMT) [185], where translation units are phrases instead of individual words. Subsequent work has introduced hierarchical phrases that can be learned from parallel texts [65, 66]. While the hierarchy need not correspond to linguistic trees, it can be seen as a simple form of syntax-based machine translation. Other studies have incorporated syntactic features in PBMT systems [38, 259]. Many other approaches to syntax-based statistical machine translation have been proposed in the literature, such as string-to-tree, tree-to-string, and tree-to-tree approaches; see [338] for a recent introduction. Another line of work considers the use of structural semantic information in machine translation, for example semantic roles [24, 119, 214, 341] and predicate-argument relations [189, 205, 343, 346]

Adequate mechanical translation can be based only on adequate structural descriptions of the languages involved and on an adequate statement of equivalences.
— V. H. Yngve

In contrast to much of the preceding line of work, neural machine translation systems are typically trained only on example translations, that is, in a string-to-string setup. While several recent studies attempted to incorporate syntax in neural machine translation in different ways [4, 63, 104, 314, 342], it is not yet clear if structural information is needed for obtaining high-quality neural machine translation systems. This section brings a different perspective to this issue by answering the following questions:

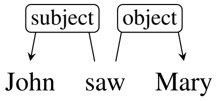
1. Do neural machine translation models acquire structural information while they are being trained on plain translations? What kind of syntactic and semantic structure is captured by these models?

2. What parts of the neural machine translation models capture more syntactic and semantic information? Do higher layers learn better representations for these kinds of properties than lower layers?



To answer these questions, I investigate the quality of neural machine translation models from the perspective of syntactic and semantic dependencies. In dependency grammar, sentence structure is represented by a labeled directed graph whose vertices are words and whose edges are relations, or dependencies, between the words [233, 260].[1] A dependency is a directed bi-lexical relation between a a head and its dependent, or modifier.

Dependency grammar has a long history. With roots in Antiquity and through Medieval times, many dependency grammar formalisms have been developed in the 20th century. Dependency syntax is typically contrasted with constituency syntax, which has been extremely influential in natural language processing (NLP). Various advantages and shortcomings are attributed to both these approaches. Dependency grammars are less expressive than constituency grammars, but they offer a better link between syntax and semantics. On the other hand, some constructions are difficult to represent in dependency formalisms (coordination is a prime example).



A dependency tree

It is not my intension to take a stand on the dependency-constituency debate. For our purposes, dependencies are attractive to study for three main reasons. First, dependency formalisms have become increasingly popular in NLP in recent years, and much work has been devoted to developing large annotated datasets for these formalisms. The Univer-

[1] The dependency graph may be defined over other lexical units than words, depending on the framework. For simplicity, this exposition will refer to words.

sal Dependencies dataset that is used in this chapter has been especially influential [261]. Second, there is a fairly rich history of using dependency structures in machine translation, although much work has focused on using constituency structures [338]. Third, as dependencies are bi-lexical relations between words, it is straightforward to obtain representations for them from a neural machine translation model. This makes them amenable to the general methodology followed in this thesis. That said, studying neural machine translation from the perspective of constituency structures is certainly a valuable venue for future work.

In this chapter, I evaluate the quality of representations from neural machine translation models for predicting syntactic and semantic dependencies, in multiple languages. I also compare with results on predicting morphological tags. The experiments on multiple languages, datasets, and models lead to the following insights:

- Morphological properties are represented sufficiently well in the lower layers of the neural machine translation model, and do not benefit from higher layers. This result is in line with the findings in Chapter 2.

- Both syntactic and semantic dependencies are better represented in the higher layers of the model. Each layer brings additional substantial improvements in representation quality.

- Higher-layers are especially helpful with predicting looser, more global, long-range dependencies such as clause-level syntactic dependencies, or second and third semantic arguments. In contrast, local, short-range dependencies do not benefit much from higher layers.

4.2 Related Work

There has been a long and rich history of using syntactic information in machine translation. There are three main paradigms that differ by where they utilize syntactic trees. String-to-tree approaches map a source sentence to a target tree, and have proved to be quite successful [53, 114, 115, 305, 347, 348]; tree-to-string approaches map a source tree to a target sentence [150, 213, 216, 256, 257, 280]; and tree-to-tree map from a source tree to a target tree [91, 100, 307, 355]. See [338] for a comprehensive introduction.

Semantic information has also been used to improve machine translation. Successful features include semantic roles [14, 24, 25, 119, 214, 339–341] and predicate-argument relations [189, 205, 343, 346]. Full semantic structures have also been considered [166].

Inspired by the use of syntax in earlier studies, recent work has started exploring how to incorporate syntax in neural machine translation. Syntactic trees may be added in different ways on the source side, in tree-to-string neural machine translation [63, 104], or on the target side, in string-to-tree translation [4, 314, 342]. Syntactic structures may also be learned jointly with the translation task [105, 144].

In terms of analysis, the most relevant work is by Shi et al. [306], who analyzed neural machine translation on different syntactic properties. They studied word-level properties (part-of-speech (POS) tags and the smallest constituent phrase above each word) and sentence-level properties (voice, tense, and top level sequence of the constituency tree). They found that local properties are better captured in lower layers of English encoders than more global properties. This chapter studies this theme in detail. The main differences from [306] are a much more diverse set of languages and models, and the investigation of both syntactic and semantic information from the perspective of dependency structures.

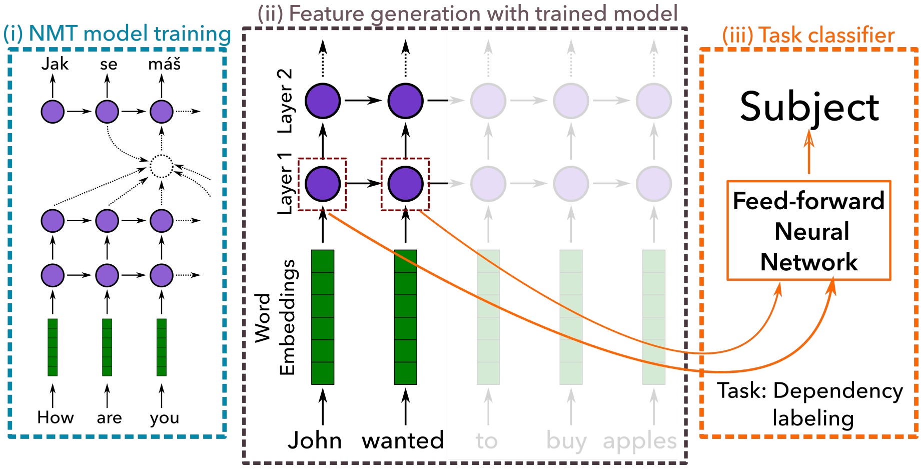
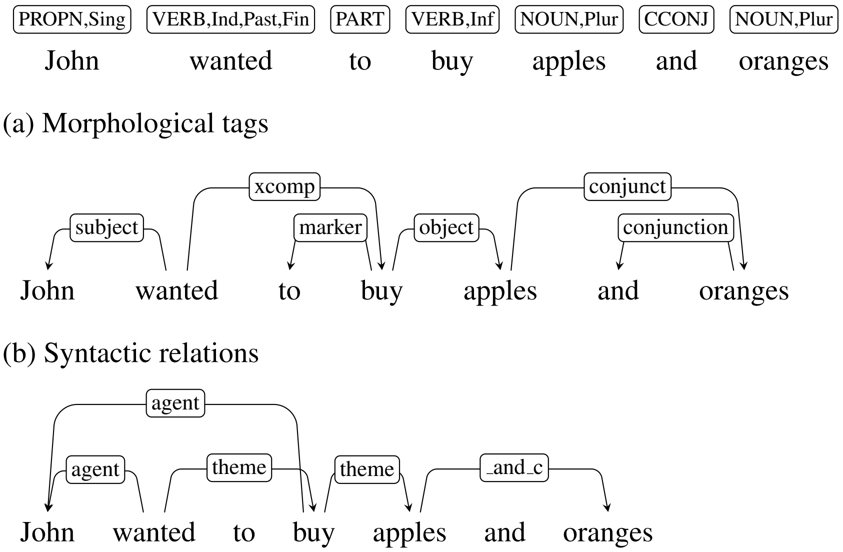


Figure 4-1: Illustration of the approach for studying syntactic and semantic relations: *(i)* a neural machine translation system trained on parallel data; *(ii)* features are generated with the pre-trained model; *(iii)* a classifier is trained on a concatenation of the generated features for the two words participating in the relation. Here a classifiers is trained on syntactic dependency labeling using features from the first encoding layer.

4.3 Methodology

The approach for evaluating relations in neural machine translation representations is similar to that used in Chapters 2 and 3. At the first step, a neural machine translation system is trained on a corpus of parallel sentences. The trained model is then used for generating word representations for every word in a given sentence. Given two words that are known to participate in a relation, a classifier is trained to predict the relation type. The input to the classifier is a concatenation of the two word representations. See Figure 4-1 for an illustration of the approach. This formulation can be seen as a dependency labeling problem, where dependency labels are predicted independently. While limited in scope, this formulation captures a basic notion of structural relations between words.[2]

[2] It is also not unrealistic, as dependency parsers often work in two stages, first predicting an unlabeled dependency tree, and then labeling its edges [229, 230]. More complicated formulations can be conceived, from predicting the existence of dependencies independently to solving the full parsing task, but dependency labeling is a simple basic task to begin with.



(a) Morphological tags

(b) Syntactic relations

(c) Semantic relations

Figure 4-2: An example sentence with different annotation schemes. **(a)** Morphological tags apply to individual words (*John* is a singular proper noun, *wanted* is a past tense, indicative, finite verb, etc.). **(b)** Syntactic relations convey dependencies between two words on a syntactic level (*John* is the subject of *wanted*, while *apples* is the object of *buy*). Every word modifies exactly one other word (it has a single incoming arc). The complete set of syntactic dependencies covers all sentence words and forms a tree. **(c)** Semantic dependencies convey predicate-argument relations between two words on a semantic level (*John* is the agent of the predicate *wanted*, but also of the predicate *buy*). The same argument word may modify two predicates (having two incoming arcs) and semantically-vacuous words do not participate in relations (*to* and *and*).

Figure 4-2 shows an example sentence, annotated with syntactic and semantic dependencies, as well as morphological tags. In the dependency labeling problem defined here, given every two words participating in a relation, the classifier predicts the relation type (edge label). For instance, given the words *John* and *wanted*, a classifier trained on syntactic dependencies needs to predict the relation `subject`. The figure also demonstrates that syntactic and semantic relations capture different structures. While *John* is the subject of *wanted*, it has no syntactic relation with the embedded verb *buy* (Figure 4-2b). In contrast, as *John* is the predicate of both *wanted* and *buy*, it has an agent relation with both of these arguments (Figure 4-2c).

The neural machine translation architecture is identical to that used in Section 3, that is, a 4-layer bidirectional recurrent neural network (RNN) encoder-decoder model with attention.[3] The classifier is the same as used in Sections 2 and 3, that is, a one-hidden layer neural network.[4] For the relation labeling task, the input to the classifier is a concatenation of encoder representations for two words in a relation, $\boldsymbol{h}_i^{S,l}$ and $\boldsymbol{h}_j^{S,l}$, where (w_i, w_j) is a known dependency with head w_i and modifier w_j.[5] The output of the classifier is a posterior distribution over the label set. For comparison purposes, experiments on morphological tagging are conducted here on a comparable dataset.

4.4 Data

The experiments in this section are conducted on six different languages: Arabic, Chinese, English, French, Russian, and Spanish. These represent diverse language families, and have the advantage of being well represented in the United Nations corpus

MT data The data set used for training the machine translation systems is the taken from United Nations proceedings [362]. As in Section 3.4, the models are trained on the first 2 million sentences of the training set. Separate models in both directions are trained for Arabic-English, Chinese-English, French-English, Russian-English, and Spanish-English, as well as an English-English autoencoder. This adds up to 11 language pairs, for each of them, three machine translation models are trained using different random initializations.

[3] See Sections 2.3 and 3.3 for more details on training the machine translation system.

[4] See Section 2.3 for more details on the classifier.

[5] Note that this formulation assumes that the *order* of the dependency is known.

	Sentences			Tokens/Relations		
	Train	Dev	Test	Train	Dev	Test
Arabic	6075	909	680	218K	29K	28K
Chinese	3997	500	500	95K	12K	12K
English	12543	2002	2077	192K	23K	23K
French	14553	1478	416	342K	34K	10K
Russian	3850	579	601	72K	11K	11K
Spanish	14187	1400	426	368K	36K	12K

(a)

	Train	Dev	Test
	Sentences		
All	33964	1692	1410
	Relations		
DM	301K	15K	12K
PAS	315K	15K	13K
PSD	440K	22K	18K

(b)

Table 4.1: Statistics for datasets of **(a)** morphological tags and syntactic relations, extracted from the Universal Dependencies datasets [261]; and **(b)** semantic dependencies, extracted from the semantic dependency parsing dataset [263, 264].

Annotated data The morphological tagging and syntactic relation labeling datasets are extracted from the Universal Dependencies dataset (v2.0) [261]. The texts in this dataset are mostly newspaper articles or web texts such as blogs and Wikipedia articles. For each word, the morphological tag is a concatenation of the POS tag with the morphological features. Roots and punctuation symbols are discarded. Sub-types are merged into their type. See Table 4.1a for dataset statistics.[6]

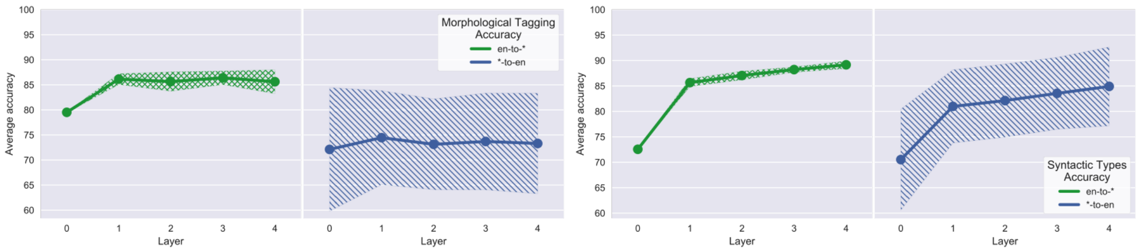
The semantic relation labeling information is extracted from the broad-coverage Semantic Dependency Parsing data set [263, 264] that includes annotations of the same set of English newspaper articles in three different semantic formalisms. Null relations are removed.[7] Table 4.1b provides some statistics on these datasets.[8]

In all cases, the classifier is trained and evaluated on the official splits to training, development, and test sets, as defined in each dataset documentation. Table C.3 in the appendix provides definitions of labels used in these datasets.

[6] More details on the datasets are available at `http://universaldependencies.org`.

[7] In practice, this means that the number of relations in each dataset is different, because of annotation differences. The number of sentences is identical (Table 4.1b).

[8] More details on the semantic formalisms are available at `http://sdp.delph-in.net`.



(a) Morphology

(b) Syntax

Figure 4-3: Results of predicting morphological tags **(a)** and syntactic relations **(b)** using representations from layers of neural machine translation systems. Representations from higher layers are more predictive than lower layers for syntactic properties, while layers from the first hidden layer are sufficient for predicting morphological properties. Layer 0 is the word embedding layer and layers 1–4 are hidden layers in the encoder neural network. The hatches show standard deviations of models trained with different initializations and language pairs.

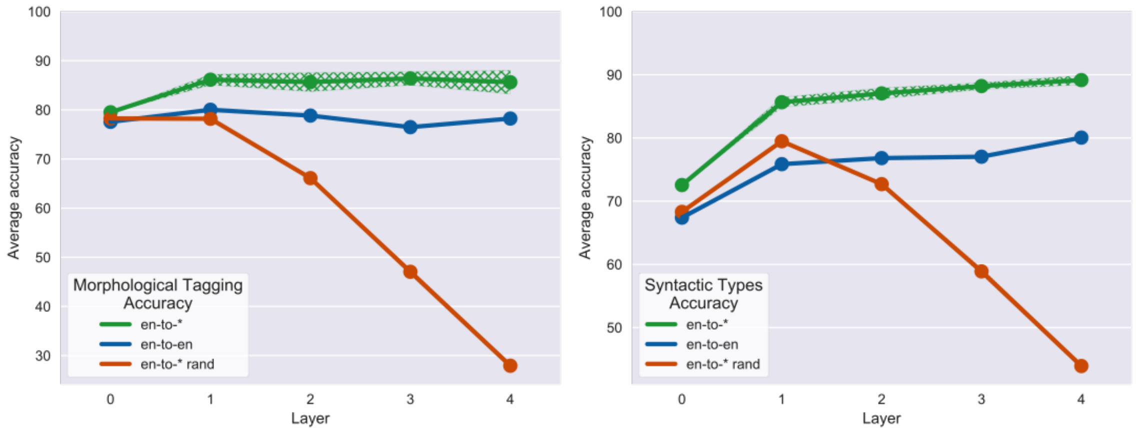
4.5 Syntactic Dependencies

Figure 4-3b shows the results of predicting syntactic dependency labels using representations from different layers in the trained models.[9] Higher layers lead to consistent and significant improvements in the quality of the representations. Representations from layer 4 perform better than representations from layer 1 in all language pairs ($p < 0.001$). Comparing successive layers, in 36/44 comparisons over 11 language pairs and 4 layer pairs (for example, layer 2 versus layer 3), the higher layer performed statistically significantly better than the lower one ($p < 0.01$).[10]

In contrast to these trends, there appears to be no benefit in using representations from higher layers to predict morphology (Figure 4-3a). In 9/11 language pairs, representations from layer 1 perform better than those from layer 4. However, only 5 of these comparisons are statistically significant ($p < 0.01$). The two cases where layer 4 representations performed better than layer 1 are not statistically significant.

[9] The results shown in the figure are averages; see Appendix C for the full results.

[10] See Section C.2 (Appendix C) for details on the statistical significance results reported in this chapter.



(a) Morphology

(b) Syntax

Figure 4-4: Results of predicting morphological tags **(a)** and syntactic relations **(b)** with representations from neural machine translation models compared to using representations from random and autoencoder models.

Once possible concern with these results is that they may be appearing because of the stacked RNN layers, and not necessarily due to the translation task. In the extreme case, perhaps even a random computation that is performed in stacked RNN layers would lead to improved performance in higher layers. This may be especially concerning when predicting relation labels, as this requires combining information about two words in the sentence. To verify that the actual translation task is important, we can look at the performance with random models, initialized in the same manner but not trained at all. Figure 4-4 shows that higher layers in random networks generally generate worse representations. In the case of morphological tagging, layers 0 and 1 are similar, but performance quickly degrades after that. When predicting syntactic dependency labels, layer 1 does improve the performance compared to layer 0. This shows that some information is captured even in random models. However, after layer 1 the performance degrades drastically, demonstrating that higher layers in random models do not generate informative representations.

The experiment with random weights shows that training the neural machine translation is important for obtaining good representations. Does the actual translation task matter? Figure 4-4 also shows the results using representations from English-English models, that is, an autoencoder scenario. As in the machine translation models, representations from higher layers do not improve morphological tagging, but do improve the prediction of syntactic dependencies. However, there is a notable degradation in representation quality when comparing the autoencoder results to those of the machine translation models. For example, the best results for predicting syntactic dependencies with the autoencoder are around 80% at layer 4. In contrast, the same layer in the translation models produces a score of 88%. In general, the representations from the machine translation models are always better than those from the autoencoder, and this gap increases as we go higher in the layers. This trend is similar to the results on morphological and semantic tagging with representations from autoencoders that were reported in the previous two chapters.[11]

4.5.1 Effect of relation type

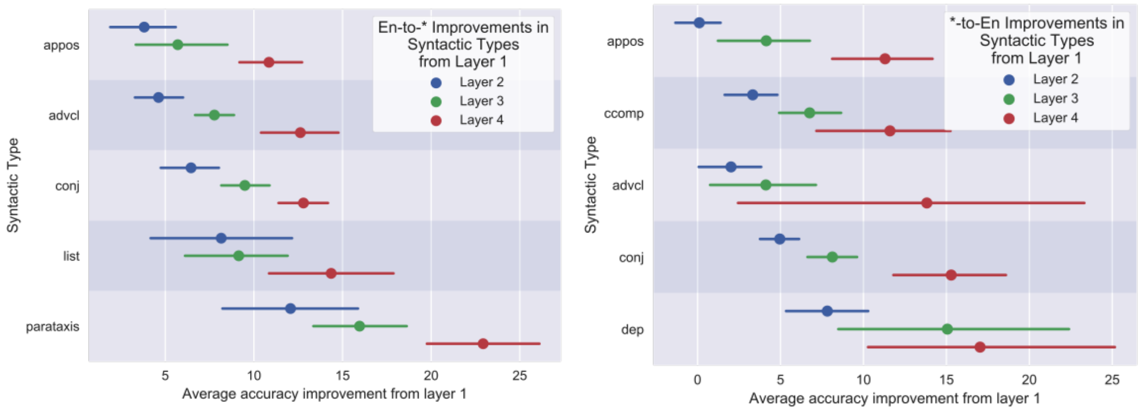
When are higher-layer representations especially important for syntactic relations? Figure 4-5 breaks down the performance according to the type of syntactic relations. The figure shows the 5 relations that benefit most from higher layer representations (see Figure C-7 in Appendix C.3 for the full results).

The general trend is that the quality of the representation improves with higher layers, with up to 20–25% improvement with representations from layer 4 compared to layer 1. The improvement is larger for certain relations:[12] dependent clauses (`advcl`, `ccomp`), loose relations (`list`, `parataxis`), and other typically long-range dependencies such as conjunctions (`conj`) and appositions (`appos`). Core nominal arguments like subject

[11] See Figure 2-5 and Table 3.2 for morphological and semantic tagging results, respectively.

[12] The list of syntactic relations in the Universal Dependencies dataset is given in Table C.3 (Appendix C.4). Refer to the online documentation for detailed definitions: `http://universaldependencies.org`.

(nsubj) and object (obj) also show consistent improvements with higher layers. Relations that do not benefit much from higher layers (Figure C-7) are mostly function words (aux, cop, det), which are local relations by nature, and the relation between a conjunct and the conjunction (cc), as opposed to the relation between two conjuncts (conj). These relations are local by nature and also typically less ambiguous. For example, the relation between a conjunction *and* and a noun is always labeled as cc, while a verb and a noun may have a subject or object relation.



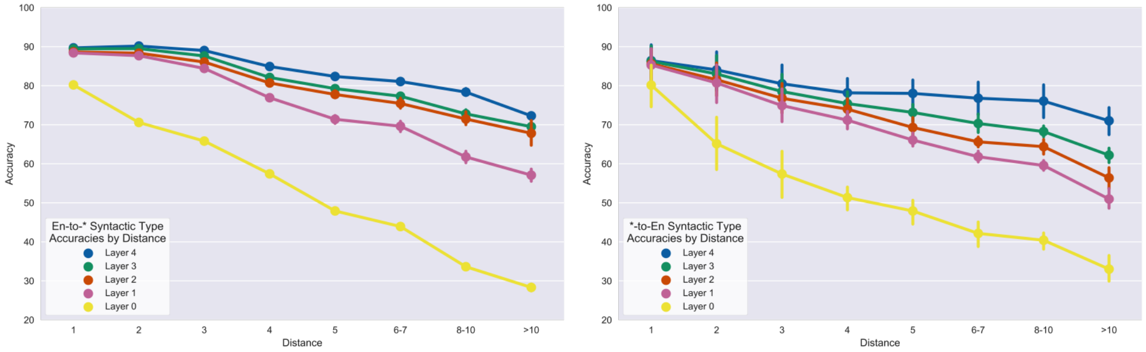
(a) Syntax, en-to-*

(b) Syntax, *-to-en

Figure 4-5: Syntactic relation types that benefit most from higher layer representations, generated by neural machine translation models trained to translate English to other languages **(a)** and other languages to English **(b)**. For the 5 relations that benefit most, the accuracy improvement is shown when using representations from layers 2/3/4 compared to layer 1.

4.5.2 Effect of relation distance

In order to quantify the notions of global and local relations, let us consider relation distance. Figure 4-6 shows the representation quality as a function of the distance between the words participating in the relation. Predicting long-distance relations is clearly more difficult than predicting short-distance ones. As the distance between the words in the relation grows, the quality of the representations decreases. When no context is available (layer 0, corresponding to word embeddings), the performance quickly drops with longer distance relations. The drop is more moderate in the hidden layers, but in low layers the effect of relation distance can still be as high as 25%. Higher layers of the network mitigate this effect and bring the decrease down to under 5%. Moreover, every layer is performing better than the previous one at each distance group. This indicates that higher layers are much better at capturing long-distance syntactic information.



(a) Syntax, en-*

(b) Syntax, *-en

Figure 4-6: Results of predicting syntactic relations at different distances between the two words participating in the relation using representations from layers of neural machine translation systems trained to translate English to other languages **(a)** and other languages to English **(b)**. Representations from higher layers are more predictive than lower layers, especially for relations with longer distances. Error bars correspond to standard deviations using models trained with different initializations and language pairs.

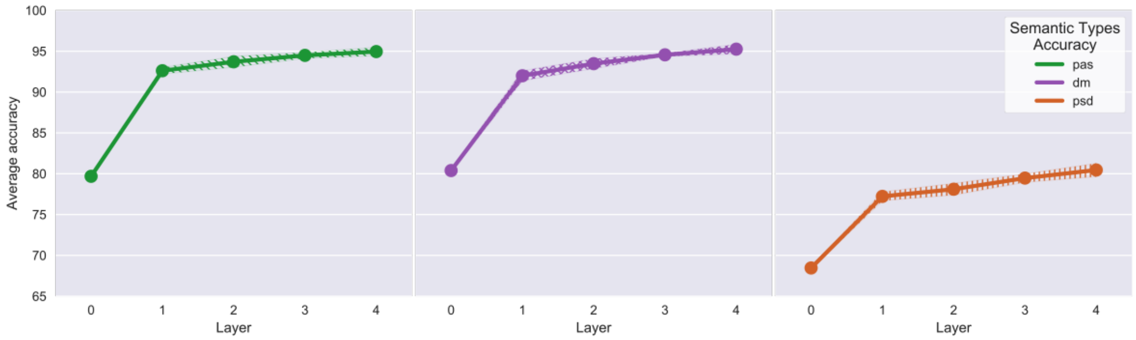
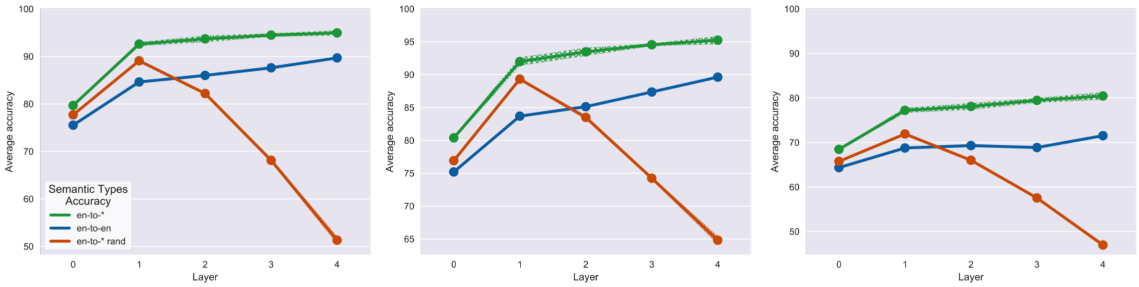


Figure 4-7: Results of predicting semantic dependencies using representations from layers of neural machine translation systems. Representations from higher layers are more predictive than lower layers for semantic properties. Layer 0 is the word embedding layer and layers 1–4 are hidden layers in the encoder neural network. The hatches show standard deviations of models trained with different initializations and language pairs.

4.6 Semantic Dependencies

Semantic dependencies exhibit similar trends to syntactic dependencies (Figure 4-7). For all language pairs and three different semantic formalisms, representations from layer 4 predict semantic relations better than those from layer 1 ($p < 0.001$). Comparing successive layers, in 59/72 comparisons over 6 language pairs, 3 semantic formalisms, and 4 layer pairs, the higher layer performed statistically significantly better than the lower one ($p < 0.01$). This shows that each successive layer brings additional improvements in representation quality for predicting both syntactic and semantic information, culminating in the top hidden layer being always better than the first hidden layer.

Considering random models, representations from layer 1 perform better than layer 0, indicating that random weights can capture some contextual information that is helpful for predicting semantic dependencies (Figure 4-8). However, performance drops rapidly after that, similarly to syntactic dependencies. Learning to translate is important for obtaining good representations, as the performance of representations from an autoencoder model is much lower. Again, this is similar to the syntactic dependencies case. These trends are consistent in all three semantic dependency formalisms.



(a) PAS (b) DM (c) PSD

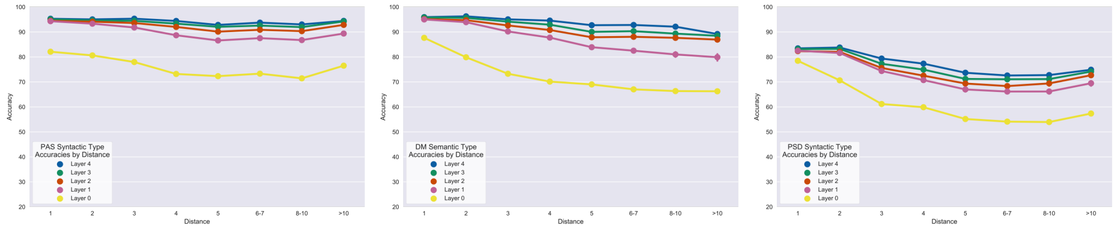
Figure 4-8: Results of predicting semantic relations with representations from neural machine translation models compared to using representations from random and autoencoder models. Results are shown on three semantic formalisms: **(a)** PAS, **(b)** DM, and **(c)** PSD.

4.6.1 Effect of relation type

Considering specific semantic relations, higher layers improve the representation quality especially in looser semantic relations such as conjunctions (Figures C-9, C-8, and C-10, in Appendix C.3). Of the core semantic arguments, `ARG3` benefits from higher layers more than `ARG2`, which in turns benefits more than `ARG1`. Thus relations that are less fundamental to the predicate benefit more from higher layer representations. The cases where higher layers do not yield much improvement are with more local relations such as numbers (`times`) and multi-word expressions (`mwe`). Note that these trends are consistent in different language pairs (small error bars) and three semantic annotation schemes.

4.6.2 Effect of relation distance

Semantic dependencies are also influenced by relation distance, similar to syntactic dependencies (Figure 4-9). It is harder to predict long-distance than short-distance relations. Lower layers degrade rapidly with long-distance relations (10–20%), while higher layers suffer much less ($< 5\%$). As before, each layer performs better than the one below it, at every distance. Therefore, higher layers are much better at capturing long-distance semantic information. These trends are consistent in all three different semantic formalisms, although the decrease in the PAS scheme is a bit milder.



(a) Sem, PAS (b) Sem, DM (c) Sem, PSD

Figure 4-9: Results of predicting semantic relations at different distances between the two words participating in the relation using representations from layers of neural machine translation systems. Representations from higher layers are more predictive than lower layers, especially for relations with longer distances. Error bars correspond to standard deviations using models trained with different initializations and language pairs. Results are shown on three semantic formalisms: PAS **(a)**, DM **(b)**, and PSD **(c)**.

4.7 Conclusion and Future Work

In this chapter, I investigated neural machine translation from the point of view of syntactic and semantic dependencies. The experiments demonstrated that higher layers generate much better representations for these properties than lower layers, especially with more global and longer-distance relations. This result is in striking contrast to morphological information that is represented better or sufficiently well in lower layers.

The notion of sentence structure explored here is quite limited. I have considered relations between words in isolation, and have only looked at labeling the relations. This can be extended in several directions. First, it will be interesting to identify the existence of a relation, either independently or by considering other relations. While this could amount to performing the full dependency parsing task, which is not trivial, lessons may be learned from recent work which attempted to jointly learn parsing and translation [105, 144, 316].

Another interesting question is how syntactic and semantic information on the target language is captured in the decoder. In Chapter 2.6, it turned out that the decoder learns very poor representations for morphology compared to the encoder. This has led to useful ideas on how to improve the neural machine translation system. Would a similar picture arise with syntax and semantics on the target side? In order to investigate this, one would need an annotated dataset of the target side of a parallel corpus. With progress in syntactic parsing, it may be possible to obtain automatic annotations from state-of-the-art parsers.

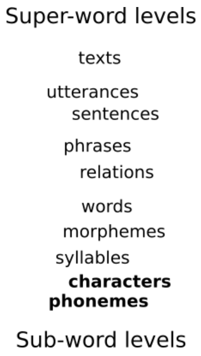
Finally, the investigation has been limited to lexical dependencies, mainly due to the methodological approach. Studying the neural machine translation representations on other syntactic and semantic formalisms would require developing a different methodology that can abstract away from the lexical items.

Chapter 5

End-to-End Automatic Speech Recognition: A Phonetic Analysis

5.1 Introduction

Traditional ASR systems are composed of multiple components, including an acoustic model, a language model, a lexicon, and possibly other components. Each of these is trained independently and combined during decoding. As such, the system is not directly trained on the speech recognition task from start to end. In contrast, end-to-end ASR systems aim to map acoustic features directly to text (words or characters). Such models have recently become popular in the ASR community thanks to their simple and elegant architecture [59, 70, 130, 234]. Given sufficient training data, they also perform fairly well. Importantly, such models do not receive explicit phonetic supervision, in contrast to traditional systems that typically rely on an acoustic model trained to predict phonetic units (e.g., HMM phone states). Intuitively, though, end-to-end models have to generate some internal representation that allows them to abstract over phonological units. For instance, a model that needs to generate the word “bought” should learn that in this case “g” is not pronounced as the phoneme /g/.



This chapter investigates if and to what extent end-to-end models *implicitly* learn phonetic representations. The hypothesis is that such models need to create and exploit internal representations that correspond to phonetic units in order to perform well on the recognition task. The linguistic units under study are phonemes and their interaction with characters.

Given a pre-trained end-to-end ASR system, I use it to generate frame-level feature representations for an acoustic speech signal. For example, these may be the hidden representations of a recurrent neural network (RNN) in the end-to-end system. I then feed these features to a classifier that is trained to predict a phonetic property of interest such as phone recognition. The performance of the classifier is used as a measure of the quality of the input features, and by proxy the quality of the original end-to-end ASR system.

This chapter aims to provide quantitative answers to the following questions:

1. To what extent do end-to-end ASR systems learn phonetic information?

2. Which components of the system capture more phonetic information?

3. Do more complicated models learn better representations for phonology? And is ASR performance correlated with the quality of the learned representations?

Two main types of end-to-end models for speech recognition have been proposed in the literature: connectionist temporal classification (CTC) [130, 234] and sequence-to-sequence learning [59, 70]. I focus here on CTC and leave exploration of the sequence-to-sequence model for future work.

To evaluate representation quality, I use TIMIT [120], a phone-segmented dataset for the phone recognition task. TIMIT comes with human-annotated time segmentation, which allows for accurate mapping between speech frames and phone labels.[1] I define a frame classification task: given representations from the CTC model, we need to classify each frame into a corresponding phone label. More complicated tasks can be conceived of—for example predicting a single phone given all of its aligned frames—but classifying frames is a basic and important task to start with.

The experimental evaluation reveals that the lowest layers in a deep end-to-end model are best suited for representing phonetic information. Applying one convolution on input features improves the representation, but a second convolution greatly degrades phone classification accuracy. Some possible explanation for this behavior are mentioned. Subsequent recurrent layers initially improve the quality of the representations. However, after a certain recurrent layer performance again drops, indicating that the top layers do not preserve all the phonetic information coming from the bottom layers. Thus, higher layers appear to focus more on character sequences than phonetic information. As another form of analysis, I cluster frame representations from different layers in the deep model and visualize them in 2D. The visualization reveals a different quality of grouping in different layers, partly corresponding to the classification results.

5.2 Related Work

End-to-end models for ASR have become increasingly popular in recent years. Important studies include models based on CTC [9, 107, 130, 234] and attention-based sequence-to-sequence models [18, 59, 70]. The CTC model is based on a recurrent neural network that takes acoustic features as input and is trained to predict a symbol per each frame. Symbols are typically characters, in addition to a special blank symbol. The CTC loss then

[1] A phone is a distinct speech sound determined by actual pronunciation while a phoneme is an abstract unit that distinguishes meaning in a given language. The annotation is TIMIT based on context-independent phones.

marginalizes over all possible sequences of symbols given a transcription. The sequence-to-sequence (seq2seq) approach, on the other hand, first encodes the sequence of acoustic features into a single vector and then decodes that vector into the sequence of symbols (characters). The attention mechanism improves upon this method by conditioning on a different summary of the input sequence at each decoding step. Section 1.6 provides more details on these models and their place in the history of ASR.

While end-to-end neural network models offer an elegant and relatively simple architecture, they are often thought to be opaque and uninterpretable. Thus researchers have started investigating what such models learn during the training process. For instance, previous work evaluated neural network acoustic models on phone recognition using different acoustic features [243] or investigated how such models learn invariant representations [352] and encode linguistic features in different layers [251, 252]. Others have correlated activations of gated recurrent networks with phone boundaries in autoencoders [335] and in a text-to-speech system [345]. Recent work analyzed different speaker representations and how well they capture various properties like speaker information, word presence, word order, utterance length, and channel information [333].

Other work analyzed joint audio-visual models. For example, in a joint model of speech and lip movements [57], phoneme embeddings were shown to be closer to certain linguistic features than embeddings based on audio alone. Chrupała et al. [73] analyzed a deep recurrent model of speech and images, and found that higher layers better capture semantic information (sentence similarity, homophone disambiguation), while lower information related to form (utterance length, word presence) is represented better at intermediate layers. Alishahi et al. [5] found that phonemes are more salient in lower layers of the same audio-visual model, although they noticed a fair amount of phonological information persisting up to the top layers. Harwath and Glass [142] observed word-like units that emerge in a model trained on pairs of images and their speech descriptions.

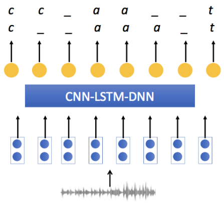
5.3 Methodology

The methodology implements the general approach (Section 1.2) in three steps. First, an end-to-end ASR system is trained on a corpus of transcribed speech. Then, the trained ASR model is used for generating frame-level feature representations on a phonetically transcribed corpus. Finally, a supervised classifier is trained on predicting frame-level phonetic outputs using the features coming from the ASR system. The classifier is evaluated on a held-out set, yielding a quantitative measure of the quality of the representations that were learned by the end-to-end ASR model.

Formally, given a sequence of acoustic features $\boldsymbol{x}$, let $\phi_t^k(\boldsymbol{x})$ denote the output of layer k of the ASR model at time t. The frame classifier takes $\phi_t^k(\boldsymbol{x})$ as input and predicts a label l_t. The rest of this section describes the ASR model and the classifier in more detail.

ASR model

The end-to-end model used in this chapter is DeepSpeech2 [9], an acoustics-to-characters system based on a deep neural network and trained with the CTC objective function (Section 1.6.2). The input to the model is a sequence of audio spectrograms (frequency log magnitudes), obtained with a 20ms Hamming window and a stride of 10ms. With a sampling rate of 16kHz, this results in 161-dimensional input features. Table 5.1a details the different layers in this model. The first two layers are convolutions where the number of output feature maps is 32 at each layer. The kernel sizes of the first and second convolutional layers are 41x11 and 21x11 respectively, where a convolution of TxF has a size T in the time domain and F in the frequency domain. Both convolutional layers have a stride of 2 in the time domain while the first layer also has a stride of 2 in the frequency domain. This setting results in 1952/1312 features per time frame after the first/second convolutional layers, respectively.



CTC-based ASR

Layer	Input Size	Output Size
cnn1	161	1952
cnn2	1952	1312
rnn1	1312	1760
rnn2	1760	1760
rnn3	1760	1760
rnn4	1760	1760
rnn5	1760	1760
rnn6	1760	1760
rnn7	1760	1760
fc	1760	29

(a) DeepSpeech2.

Layer	Input Size	Output Size
cnn1	161	1952
cnn2	1952	1312
lstm1	1312	600
lstm2	600	600
lstm3	600	600
lstm4	600	600
lstm5	600	600
fc	600	29

(b) DeepSpeech2-light.

Table 5.1: Architectures of the end-to-end ASR models used in this work, following the DeepSpeech2 models [9].

The convolutional layers are followed by 7 bidirectional recurrent layers, each with a hidden state size of 1760 dimensions. Notably, these are simple RNNs and not gated units such as long short-term memory (LSTM) [149], as this was found to produce better performance [9]. The experiments below also compare with a shallower version of the model, called DeepSpeech2-light, which has 5 layers of bidirectional LSTMs, each with 600 dimensions (Table 5.1b). This model runs faster but leads to worse recognition results.

Each convolutional or recurrent layer is followed by batch normalization [160, 193] and a rectified linear unit (ReLU) non-linearity. The final layer is a fully-connected layer that maps onto the number of symbols (29 symbols: 26 English letters plus space, apostrophe, and a blank symbol).

Supervised Classifier

The frame classifier takes features $\phi_t^k(\boldsymbol{x})$ from different layers of the DeepSpeech2 model as input and predicts a phone label. The size of the input to the classifier thus depends on which layer in DeepSpeech2 is used to generate features (see Table 5.1). The classifier is modeled as a feed-forward neural network with one hidden layer, where the size of the hidden layer is set to 500. This is followed by dropout ($\rho = 0.5$) and a ReLU non-linearity, then a Softmax layer mapping onto the label set size (the number of unique phones). This simple formulation helps focus on the quality of the representations learned by the ASR model, rather than improving the state-of-the-art on the supervised task.

The classifier is trained with Adam [179] with the recommended parameters ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = e^{-8}$) to minimize the cross-entropy loss. Training is run with a mini-batch size of 16 for 30 epochs, and the model with the best development loss is used for evaluation.

5.4 Data and Tools

The experiments utilize the `deepspeech.torch` [254] implementation, which comes with pre-trained models of both DeepSpeech2 and the simpler variant DeepSpeech2-light. The end-to-end models are trained on LibriSpeech [266], a publicly available corpus of English read speech, containing 1,000 hours sampled at 16kHz. The word error rates (WERs) of the DeepSpeech2 and DeepSpeech2-light models on the Librispeech-test-clean dataset are 12 and 15, respectively, as reported in [254].

The frame classification dataset is extracted from TIMIT [120], which comes with time segmentation of phones. The official train/development/test split is used for all experiments. Table 5.2b summarizes statistics of the extracted frame classification dataset. Note that due to sub-sampling at the DeepSpeech2 convolutional layers, the number of frames decreases by a factor of two after each convolutional layer. The possible labels are the 60 phone symbols included in TIMIT (excluding the begin/end silence symbol *h#*). Table 5.2a shows the number of frames per phone in the training set.

s	69903	ey	28743	ah	20136	ux	14026	ch	7077	en	4903
iy	44107	aa	27753	dcl	19995	w	13393	th	6830	epi	3831
ix	37676	l	27095	k	19790	v	12028	jh	6200	uh	3825
n	34397	r	26315	t	19277	bcl	11965	hh	6059	b	3770
ih	33320	kcl	24610	axr	18872	aw	11704	d	5875	g	3179
ae	31136	ao	22937	pcl	18432	p	11455	uw	5565	nx	1940
z	30832	f	22801	ax	17060	el	8609	y	5466	zh	1217
eh	30533	m	22250	pau	16711	dh	8538	dx	5367	ax-h	1188
ay	29922	ow	21179	q	16653	ng	7279	oy	5110	em	1010
tcl	29916	er	20286	sh	15571	gcl	7099	hv	5098	eng	198

(a) Number of frames per phone in our training data, extracted from the TIMIT training set.

		Train	Dev	Test
Utterances		3696	400	192
Frames	input	988K	108K	50K
	after cnn1	494K	54K	25K
	after cnn2	234K	25K	12K

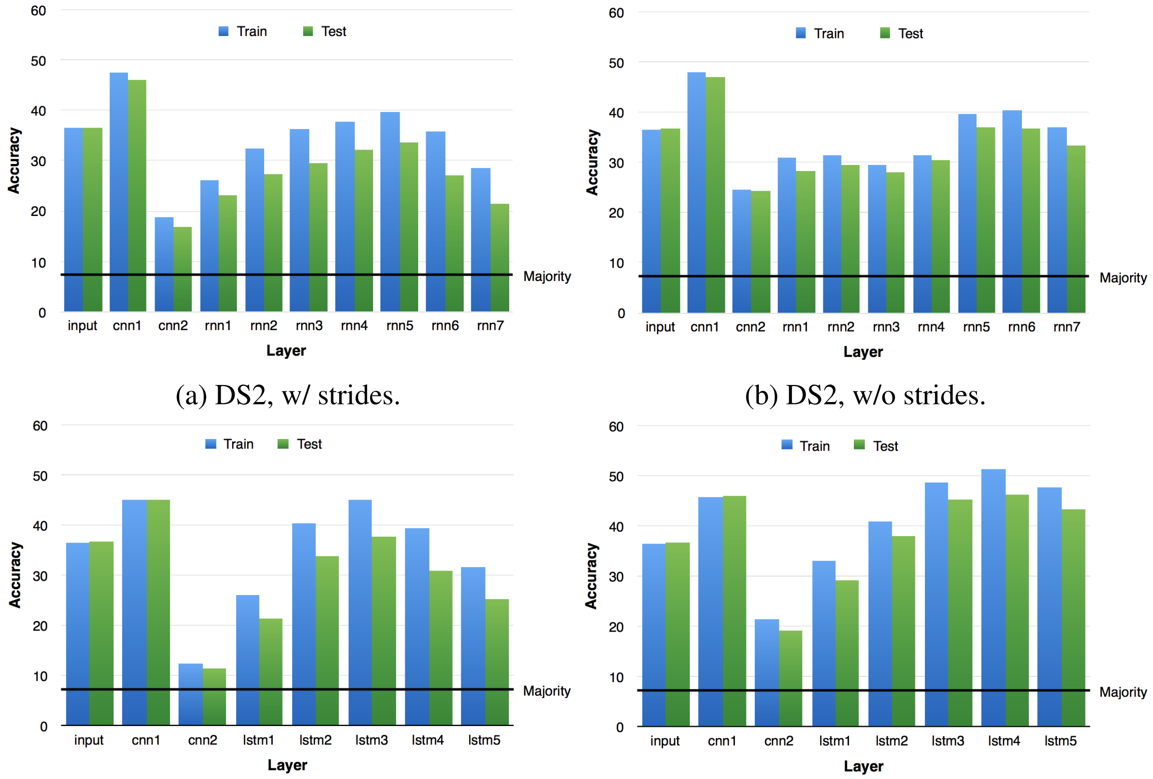
(b) Frame classification data extracted from TIMIT.

Table 5.2: Statistics of the frame classification dataset extracted from TIMIT [120]. **(a)** Number of frames per phone. **(b)** The split into training, development, and test sets.

5.5 Main Results

Figure 5-1a shows frame classification accuracy using features from different layers of the DeepSpeech2 model. The results are all above a majority baseline of 7.25% (the phone [s]). Input features (spectrograms) lead to fairly good performance, considering the 60-wise classification task. The first convolution further improves the results, in line with previous findings about convolutions as feature extractors before recurrent layers [289]. However, applying a second convolution significantly degrades accuracy. This can be attributed to the filter width and stride, which may extend across phone boundaries.[2] Nevertheless, the large drop is quite surprising.

[2] The two convolutions downsample by x4, so some time resolution may be lost.



(a) DS2, w/ strides.

(b) DS2, w/o strides.

(c) DS2-light, w/ strides.

(d) DS2-light, w/o strides.

Figure 5-1: Frame classification accuracy using representations from different layers of DeepSpeech2 (DS2) and DeepSpeech2-light (DS2-light), with or without strides in the convolutional layers.

The first few recurrent layers improve the results, but after the 5th recurrent layer accuracy goes down again. One possible explanation to this may be that higher layers in the model are more sensitive to long distance information that is needed for the speech recognition task, whereas the local information that is needed for classifying phones is better captured in lower layers. For instance, to predict a word like "bought", the model would need to model relations between different characters, which would be better captured at the top layers.[3] In contrast, feed-forward neural networks trained on phone recognition

[3] As another example, consider the possible pronunciations of the letter "c" in English: /s/ and /k/. It is

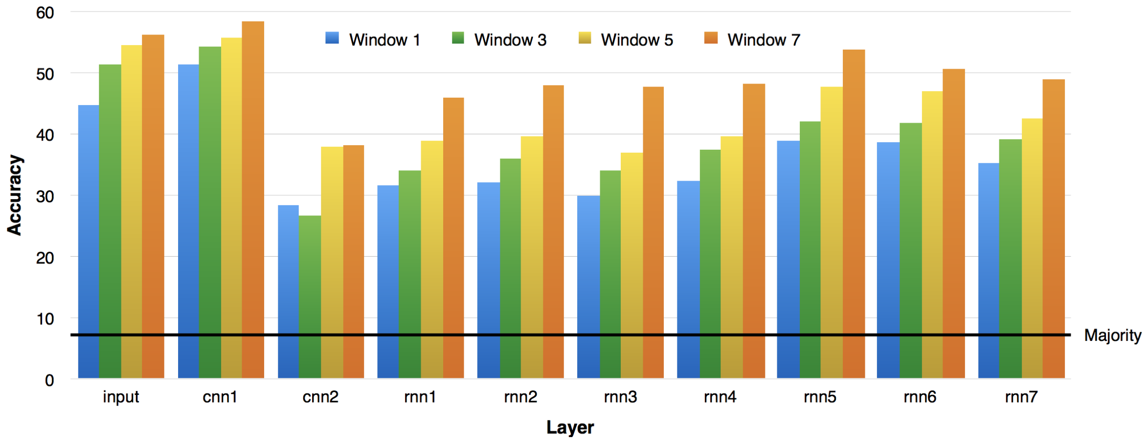


Figure 5-2: Frame classification accuracy using different window widths around the current frame.

were shown to learn increasingly better representations at higher layers [251, 252]; such networks do not need to model the full speech recognition task, different from end-to-end models.

The trends shown in Figure 5-1a are consistent in multiple configurations, including different input features, output labels, classifiers,[4] and DeepSpeech2 variants. Figure 5-1 shows results with several different network configurations. We will return to these in the next section.

For now, Figure 5-2 shows test set results with different window widths around the frame that is to be classified. This improves the representation and also accounts for possible delay effects [293]. As expected, larger windows improve the representation quality. The absolute numbers are much better than using only a single frame (+10–15%), but the overall trend for a given window size is similar: initial performance drop after

possible that in some intermediate layers it is beneficial to be able to distinguish between these pronunciation, leading to a higher classification accuracy, while the top layers may be more focused on identifying the letter "cc", since these layers are closer to the text output.

[4] A linear classifier produces accuracies lower by about 4–5% at every layer, but the relative layer-wise trends are the same. See Table D.1 in Appendix D.1.

the convolutional layers, then steady increase at the first recurrent layers and another drop at the top layers. The drop is somewhat more moderate than in the single frame case (compare to Figure 5-1b), indicating that some shifting effect may indeed be taking place, although it might be limited given that DeepSpeech2 is using bidirectional RNNs (the results in [293] are with unidirectional RNNs).

The following section investigate several aspects of the model: model complexity, effect of strides in the convolutional layers, and effect of blanks. This is followed by a discussion of classification into different output label sets. Then a visualization of frame representations in 2D provides another look at the quality of different layers.

5.6 Analysis

5.6.1 CNN strides

The original DeepSpeech2 models have convolutions with strides (steps) in the time dimension [9]. This leads to subsampling by a factor of 2 at each convolutional layer, resulting in reduced dataset size (see Table 5.2b). Consequently, the comparison between layers before and after convolutions is not entirely fair. To investigate this effect, Figure 5-1b shows the results of generating features from the DeepSpeech2 model at different layers without using strides in the convolutions.[5] The general trend is similar to the strided case: large drop at the 2nd convolutional layer, then steady increase in the recurrent layers with a drop at the final layers. However, the overall shape of the accuracy in the recurrent layers is less spiky; the initial drop is milder and performance does not degrade as much at the top layers. A similar pattern is observed in the non-strided case of DeepSpeech2-light (Figure 5-1d).

[5] Note that the model was still trained with strided convolution, but the convolutions are run without strides while generating features for the classifier.

These results can be attributed to two factors. First, running convolutions without strides maintains the number of examples available to the classifier, which means a larger training set. More importantly, however, the time resolution remains high which can be important for frame classification.

5.6.2 Recurrent layer

Figure 5-1c shows the results of using features from the DeepSpeech2-light model. This model has less recurrent layers (5 vs. 7) and smaller hidden states (600 vs. 1760), but it uses LSTMs instead of simple RNNs. A first observation is that the overall trend is the same as in DeepSpeech2: significant drop after the first convolutional layer, then initial increase followed by a drop in accuracy in the final recurrent layers.

Comparing the two models (Figures 5-1a and 5-1c), a number of additional observations can be made. First, the convolutional layers of DeepSpeech2 contain more phonetic information than those of DeepSpeech2-light (+1% and +4% for cnn1 and cnn2, respectively). In contrast, the recurrent layers in DeepSpeech2-light are better, with the best result of 37.77% in DeepSpeech2-light (by lstm3) compared to 33.67% in DeepSpeech2 (by rnn5). This suggests again that higher layers do not model phonology very well: when there are more recurrent layers, the convolutional layers compensate and generate better representations for phonology than when there are fewer recurrent layers. Interestingly, the deeper model performs better on the speech recognition task (12% WER with DeepSpeech2 compared to 15% WER with DeepSpeech2-light [254]) while its deep representations are not as good at capturing phonology, suggesting that its top layers focus more on modeling character sequences, while its lower layers focus on representing phonetic information.

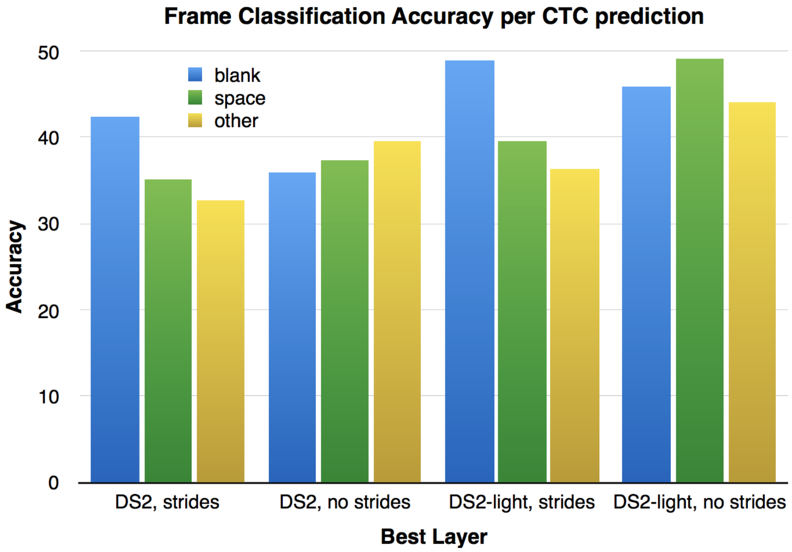


Figure 5-3: Frame classification accuracy at frames predicted as blank, space, or another letter by DeepSpeech2 and DeepSpeech2-light, with and without strides in the convolutional layers.

5.6.3 Blanks

Recall that the CTC model predicts either a letter in the alphabet, a space, or a blank symbol. This allows the model to concentrate probability mass on a few frames that are aligned to the output symbols in a series of spikes, separated by blank predictions [131]. Figure 5-3 breaks the performance down into cases where the ASR model predicted a blank, a space, or another letter. Results are shown using representations from the best recurrent layers in DeepSpeech2 and DeepSpeech2-light, run with and without strides in the convolutional layers. In the strided case, the hidden representations are of highest quality for phone classification when the model predicts a blank. This appears counterintuitive, considering the spiky behavior of CTC models, which should be more confident when predicting non-blank. However, it turns out that only 5% of the frames are predicted as blanks, due to downsampling in the strided convolutions. When the model is run without

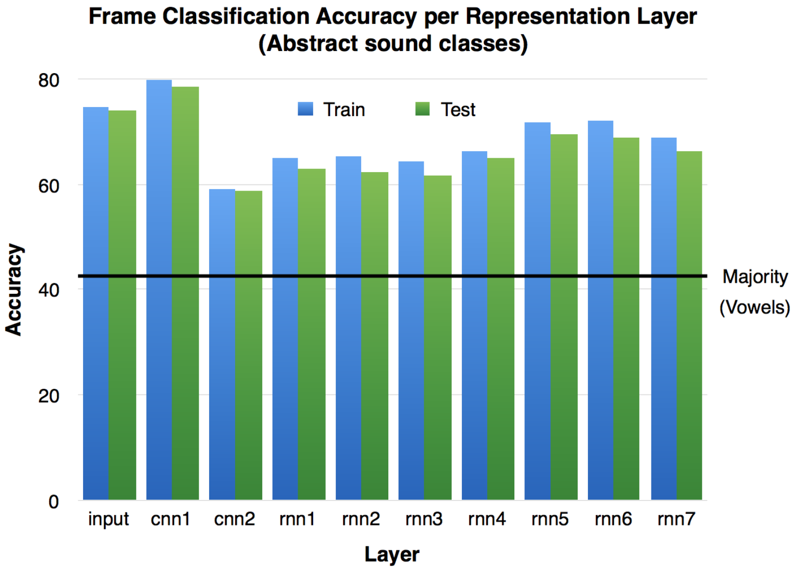


Figure 5-4: Accuracy of classification into sound classes using representations from different layers of DeepSpeech2.

strides, a somewhat different behavior appears. In this case the model predicts many more blanks (more than 50% compared to 5% in the non-strided case), and representations of frames predicted as blanks are not as good, which is more in line with the common spiky behavior of CTC models [131].

5.6.4 Output labels

The preceding experiments were conducted with a label set of 60 phones. However, speech sounds are often organized in coarse categories like consonants and vowels. This section investigates whether the ASR model learns such categories. The primary question we ask is: which parts of the model capture most information about coarse categories? Are higher layer representations more informative for this kind of abstraction above phones?

Figure 5-4 shows the results of classifying frames into the following coarse-grained

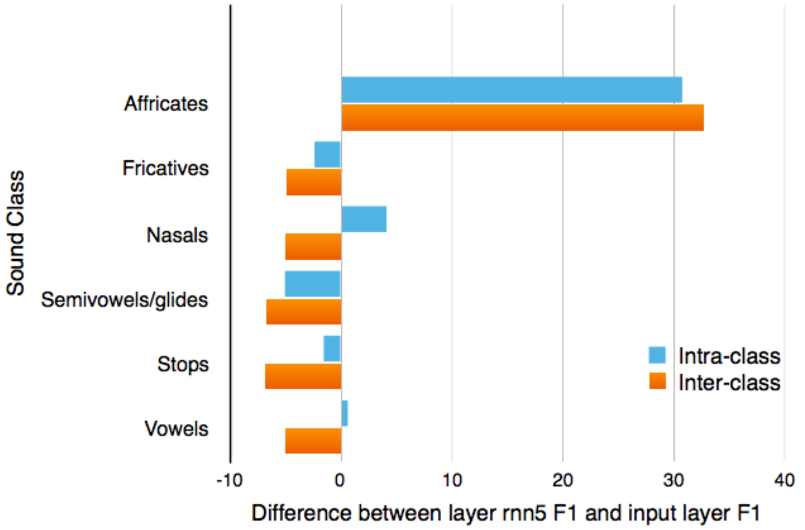
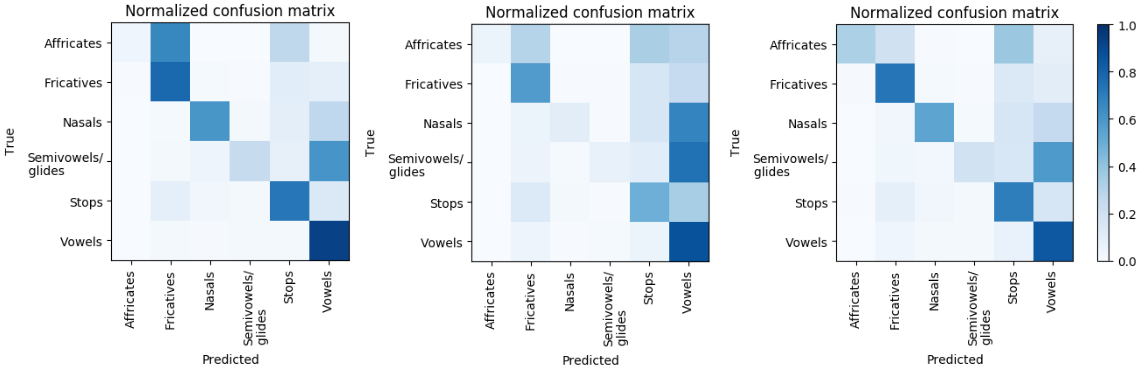


Figure 5-5: Difference in F_1 score using representations from layer rnn5 compared to the input layer, showing F_1 within each sound class ("intra-class") and among different classes ("inter-class").

categories: affricates, fricatives, nasals, semivowels/glides, stops, and vowels.[6] All layers produce representations that contain a non-trivial amount of information about sound classes (above the vowel majority baseline). As expected, predicting sound classes is easier than predicting phones, as evidenced by a much higher accuracy compared to the previous results. As in previous experiments, the lower layers of the network (input and cnn1) produce the best representations for predicting sound classes. Performance then first drops at cnn2 and increases steadily with each recurrent layer, finally decreasing at the last recurrent layer. It appears that the top layer does not generate better representations for abstract sound classes.

Let us look more closely at the difference between the input layer and the best recurrent layer (rnn5), broken down to specific sound classes. Figure 5-5 shows the change in

[6] The mapping between phones and their coarse-grained categories follows that defined in the TIMIT documentation [120].



(a) input (b) cnn2 (c) rnn5

Figure 5-6: Confusion matrices of sound class classification using representations from different layers.

F_1 score when moving from input representations to rnn5 representations, where F_1 is calculated in two ways. The *inter-class* F_1 is calculated by directly predicting coarse sound classes, thus measuring how often the model confuses two separate sound classes. The *intra-class* F_1 is obtained by predicting fine-grained phones and micro-averaging F_1 inside each coarse sound class (not counting confusion outside the class). It indicates how often the model confuses different phones in the same sound class. As Figure 5-5 shows, in most cases representations from rnn5 degrade the performance, both within and across classes. There are two notable exceptions. Affricates are better predicted at the higher layer, both compared to other sound classes and when predicting individual affricates. It may be that more contextual information is needed in order to detect a complex sound like an affricate. Second, the intra-class F_1 for nasals improves with representations from rnn5, whereas the inter-class F_1 goes down, suggesting that rnn5 is better at distinguishing between different nasals.

Figure 5-6 shows confusion matrices of predicting sound classes using representations from the input, cnn2, and rnn5 layers. Much of the confusion arises from confusing relatively similar classes: semivowels/vowels, affricates/stops, affricates/fricatives. Interestingly, affricates are less confused at layer rnn5 than in lower layers, which is consistent with our previous observation.

Finally, Figure 5-7 reports experiments with a reduced set of 48 phones [195], exhibiting a similar trend to the other label sets. Interestingly, as with sound classes, the affricates [ch] and [jh] are better represented at rnn5 (F_1 scores of 42.5% and 34.9%, respectively) than at the input layer (7.2% and 8.3%).

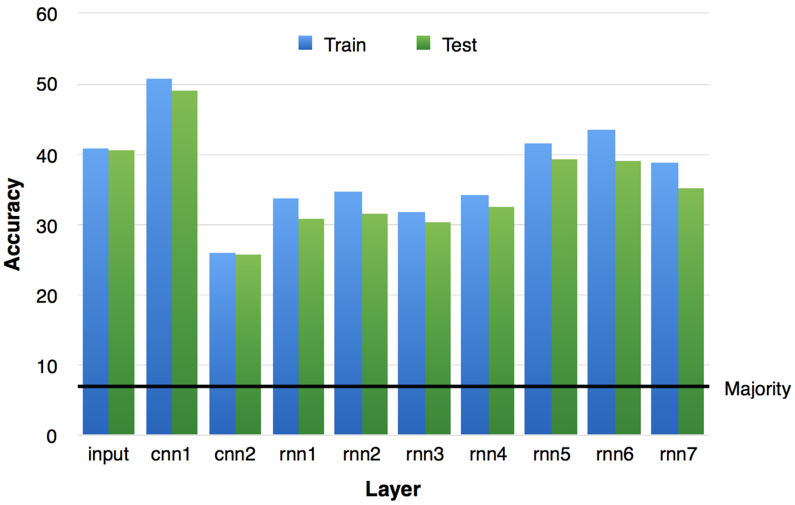


Figure 5-7: Frame classification accuracy with a reduced set of 48 phones.

5.6.5 Clustering and visualizing representations

This section concludes the experimental results with visualizations of frame representations from different layers of DeepSpeech2.[7] First, the DeepSpeech2 model was run on the entire development set of TIMIT to generate feature representations for every frame from all layers. This results in more than 100K vectors of different sizes. Then, the vectors in each layer were clustered with k-means ($k = 500$) and the cluster centroids were plotted

[7] The visualization was obtained following a similar procedure to that of [142].

using t-SNE [223]. Each cluster is assigned the phone label that had the largest number of examples in the cluster.

Figure 5-8 shows t-SNE plots of cluster centroids from selected layers, with color and shape coding for the phone labels (see Figure D-1 in Appendix D for other layers). The input layer produces clusters which show a fairly clean separation into groups of centroids with the same assigned phone. After the input layer it is less easy to detect groups, and lower layers do not show a clear structure. Layers rnn4 and rnn5 again display some meaningful groupings (e.g., [z] on the right side of the rnn5 plot), after which rnn6 and rnn7 again show less structure.

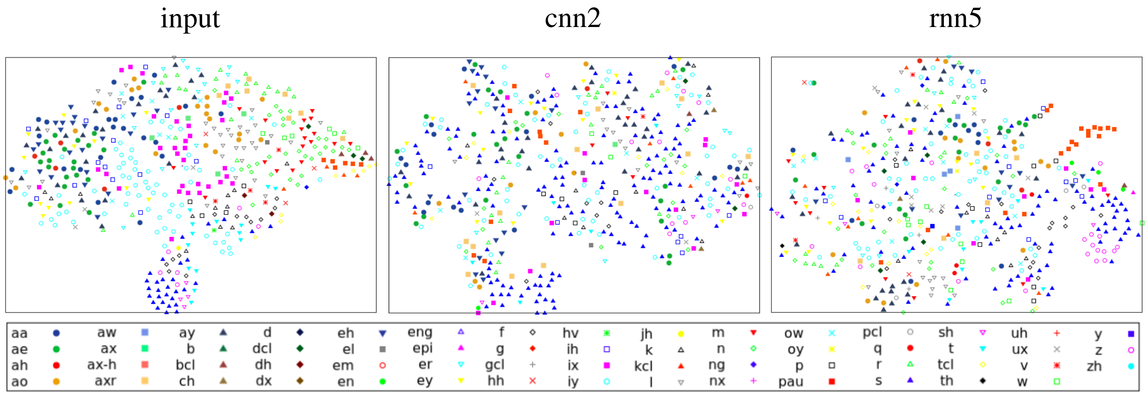


Figure 5-8: Centroids of frame representation clusters using features from different layers.

Figure D-2 (in Appendix D) shows clusters that have a majority label of at least 10–20% of the examples.[8] In this case groupings are more observable in all layers, and especially in layer rnn5.

Note that these results are mostly in line with our previous findings regarding the quality of representations from different layers. It appears that when frame representations are better separated in vector space, the classifier does a better job at classifying frames

[8] As some clusters are quite noisy, it is useful to prune clusters where the majority label does not cover enough of the cluster members, depending on the number of examples left in each cluster after pruning

into their phone labels. A similar observation was made in [252]. They found that both classification accuracy and representation separability improve in higher layers of a neural network trained on phone recognition. Interestingly, in their case performance does not drop at higher layers. The reason for the difference with the results reported here may be that their model is trained on phone recognition, and thus the auxiliary classification task is aligned with the original training objective. In contrast, the end-to-end model was trained on predicting characters, and so its representations at the top layer are better tuned to this property, whereas phonetic discrimination is important only as an intermediate step.

5.7 Conclusion and Future Work

In this chapter, I analyzed representations in a deep end-to-end ASR model that is trained with a CTC loss. I empirically evaluated the quality of the representations on a frame classification task, where each frame is classified into its corresponding phone label. I compared feature representations from different layers of the ASR model and observed striking differences in their quality. Interestingly, intermediate layers capture phonetic information better than the top layer. This can be explained by the end-goal of the ASR model, which is trained on predicting character sequences in an end-to-end manner, different from traditional acoustic model. In addition, visualizations demonstrate that differences in classification accuracy in different layers may correspond to the separability of the representations in vector space.

Future work can extend this analysis to other speech features, such as syllable structure, speaker identification and verification, and dialect or language identification. Experimenting with other end-to-end systems, such as sequence-to-sequence (seq2seq) models and acoustics-to-words systems, is another interesting direction.

Another venue for future work is to improve the end-to-end model based on the results of this analysis, for example by improving the representation capacity of certain layers in the deep neural network. Understanding representation learning at different layers of the end-to-end model can guide joint learning of phone recognition and ASR, as recently proposed in a multi-task learning framework [320].

Chapter 6

Afterword

This work was concerned with understanding the internal representations learned by language and speech processing models. We started with a general methodology for conducting informed deep learning research, where quantitative analysis guides the research process. The body of work presented in this thesis is focused on the analysis part: what linguistic information is captured by end-to-end neural networks when they are trained on large amounts of data, where and how is this information represented, and what is the interplay between different parts of the neural network. I studied these themes in the context of two fundamental language technology tasks and through the lens of core language properties. Chapter 2 investigated morphology in neural machine translation and found that morphological information is better represented at lower layers of the neural machine translation model. Chapter 3 contrasted part-of-speech and semantic tagging, and found that lexical semantic information tends to be captured more in the higher layers of the models. Chapter 4 took one step up the language hierarchy, and evaluated syntactic and semantic relations in different layers. The combined results of these three chapters suggest a hierarchical organization of linguistic information in neural networks that are trained on the machine translation task. Lower layers of the network tend to focus on simple, local properties, while higher layers focus on more complex, global properties.

Chapter 5 extended the analysis to automatic speech recognition, and found that phonetic information is represented better in intermediate layers of a deep end-to-end model than the top layers. This suggested that higher layers in the model are more concerned with abstracting over phonetic distinctions and capturing character patterns. This too can be seen as a notion of emerging hierarchy.

In closing, I would like to offer several directions for future study.

Linguistic properties The models and tasks investigated in this thesis are definitely not the whole story. Language has more complex structures that deserve their own study. Moving beyond relations into phrase structure, sentence structure, and beyond is one possible direction to explore. Do neural machine translation models learn such properties? Our recent work suggests that neural machine translation representations fail to represent certain semantic properties [275], but more research is needed on this topic.

The speech recognition experiments were limited to a very basic property: classifying speech frames. Do end-to-end models learn more complex units, such as syllables, words, and beyond? What about speaker, dialect or language information? Investigating end-to-end models from these perspectives would shed more light on how they work.

Models and architectures Another natural extension is to investigate other end-to-end models. Within machine translation and speech recognition, new architectures are proposed every day. Do these capture language in a similar manner to the standard end-to-end models that were investigated here? Can we make more informed choices of model architecture and components by analyzing their internal representations? And what about end-to-end models for other language processing tasks?

On the other side, one may also consider simplified versions of neural network models whose behavior is better understood. For instance, constructing small-scale models trained on synthetic datasets can lead to a more complete analysis [155, 328]. Working with synthetic data can also help verify that the methodology works as expected, by constructing a dataset with some known underlying property and training a classifier to uncover it.

Methods This thesis followed a unified methodology for analyzing deep learning models for language and speech processing. It has proven quite useful in leading to interesting insights regarding the internal representations in such models. However, this methodology has its limitation. First, training a classifier to predict certain properties is an indirect way to measure association between neural network representations and linguistic properties. Forming more direct links might shed a different light on the questions we ask. One possibility is to frame the problem in information theoretic terms, and measure properties like mutual information between internal representations and target properties. An intriguing question is how to track information flow inside the model and observe how some information is lost, as we have seen that some kinds of linguistic information are lost in higher layers. Note that such an information theoretic approach would have to somehow handle the high-dimensional space of the distributed representations learned by neural networks. Another interesting direction is to investigate *causal* relationships between internal representations and linguistic properties. Do they end-to-end models have to learn linguistic representations to perform well on their tasks?

Second, the analysis in this thesis provides *global* results, at the model-level or at the level of model components. The results do not provide direct explanations for specific, local model predictions. Generating such explanations for automatic predictions is arguably important [94, 95], and some related work in language processing attempts to go in this direction (see Section 1.3.2). But this is only the beginning; there is room for much more work in this area.

Lastly, evaluation of interpretation methods remains challenging. Ultimately, the results need to be evaluated by humans on some real task [94], but this is not trivial to accomplish. Using human behavioral experiments may be a reasonable proxy [60, 113, 249].

Closing the loop In the introduction to this thesis, I argued that one important outcome of the analysis should be insights for improving the original end-to-end system. We have seen one example for this, where our analysis of morphology in the neural machine translation encoder and decoder led us to try and improve morphological learning in the decoder (Section 2.7). Multi-task learning turned out to be a powerful technique in this case. I believe that other results in this thesis can suggest directions for closing the loop and improving the original models. For instance, it may be beneficial to use auxiliary loss functions at different layers, as projected from our analysis. Indeed, recent work has picked up on the idea that different layers capture different linguistic properties, exploiting this to generate better contextualized word representations [274].

Between humans and machines In conclusion, I allow myself a bit of speculation. This thesis studied how machines—certain artificial neural networks—learn language. But the most successful language learning machine is obviously humans. Despite their great success, deep learning models remain limited and lag behind human performance on many tasks. Can we learn something from how humans learn and process language that would help us develop better machines? Past advances were inspired by how humans process information. Known examples are CNN architectures that are inspired by the human visual processing system, and the speech feature representations like MFCCs that are inspired by the human auditory processing system. At present, there is still much unknown about how humans process and produce language, but future advances might expose our amazing language processing system in a way that is beneficial for developing better artificial systems.

There also is some reason to hope that insights from machine learning can help guide the investigation of human language processing in psycholinguistic and neurolinguistic research. The emerging hierarchical structure in deep learning models of language is one interesting place to look at. Without more direct evidence, one cannot claim that humans *must* process language with similar mechanisms. But the analysis of artificial neural networks might tell us something about how humans *might* be processing language.

Appendix A

Morphological Tagging Experiments

A.1 Additional results for the effect of target language

Section 2.5.3 investigated the effect of the target language on source-side representations. Table A.1 shows additional part-of-speech (POS) tagging results in German and Czech, confirming that translating into a simpler language (English) results in better source-side representations. As before, the autoencoder model learns much worse representations.

	Target		
Source	English	Arabic	Self
German	93.5	92.7	89.3
Czech	75.7	75.2	71.8

Table A.1: POS tagging accuracy in German and Czech when translating into different target languages. Self = German/Czech in rows 1/2 respectively.

Appendix B

Semantic Tagging Experiments

B.1 Full results for coarse-grained semantic tagging

Table B.1 shows semantic (SEM) tagging results with coarse-grained tags, using representations from different layers. All pairwise comparisons between two layers are statistically significant at $p < 0.001$ except for layer 3 vs. layer 4 with an Arabic target language (significant at $p < 0.01$) and layer 2 vs. layer 3 with a Russian target language (not significant). Statistical significance was calculated by the approximate randomization test [265].

	Arabic	Spanish	French	Russian	Chinese
0	85.7	85.7	85.7	85.8	85.6
1	90.7	90.5	90.6	90.6	90.5
2	90.3	90.4	90.3	90.3	90.0
3	90.6	90.7	90.8	90.2	90.3
4	91.1	91.3	91.2	91.0	90.7

Table B.1: SEM tagging accuracy on English with coarse-grained tags using features generated by different encoding layers of 4-layered neural machine translation models trained with different target languages.

B.2 Statistical significance

Table B.2 shows statistical significance results (calculated according to [265]) when comparing representations generated by models trained with different target languages. Each cell shows significance for a comparison of classifiers trained on representations from models trained with two different target languages.

	Ar-Es	Ar-Fr	Ar-Ru	Ar-Zh	Es-Fr	Es-Ru	Es-Zh	Fr-Ru	Fr-Zh	Ru-Zh
					Fine-grained SEM tags					
0	ns	ns	ns	‡	ns	ns	‡	ns	*	‡
1	‡	*	ns	‡	†	‡	ns	†	*	‡
2	†	ns	ns	‡	‡	‡	‡	ns	‡	‡
3	†	†	‡	‡	ns	‡	‡	‡	‡	ns
4	‡	*	‡	‡	‡	‡	‡	‡	‡	‡
					Coarse-grained SEM tags					
0	†	*	ns	‡	ns	‡	ns	†	*	‡
1	‡	*	*	‡	†	‡	ns	ns	*	*
2	ns	ns	ns	‡	ns	†	‡	ns	‡	‡
3	‡	‡	‡	‡	ns	‡	‡	‡	‡	*
4	‡	ns	‡	‡	†	‡	‡	‡	‡	‡
					POS tags					
0	*	*	†	‡	ns	ns	*	ns	ns	ns
1	‡	‡	‡	‡	ns	*	‡	ns	‡	‡
2	ns	ns	ns	‡	ns	ns	‡	ns	‡	‡
3	‡	*	†	‡	†	‡	‡	‡	‡	‡
4	‡	‡	ns	‡	ns	‡	‡	‡	‡	‡

Table B.2: Statistical significance results for SEM and POS tagging experiments (Chapter 3) comparing different target languages: Arabic (Ar), Spanish (Es), French (Fr), Russian (Ru), and Chinese (Zh). ns = $p > 0.05$, * = $p < 0.05$, †= $p < 0.01$, ‡= $p < 0.001$.

Appendix C

Relation Prediction Experiments

C.1 Full results

This section provides detailed results to complement Chapter 4. Figure C-1 shows the results of predicting morphological tags with representations from encoders of all the different neural machine translation model. Figure C-2 shows similar results for the syntactic dependency labeling tasks, and Figures C-3, C-4, and C-5 show the results for the three semantic formalisms.

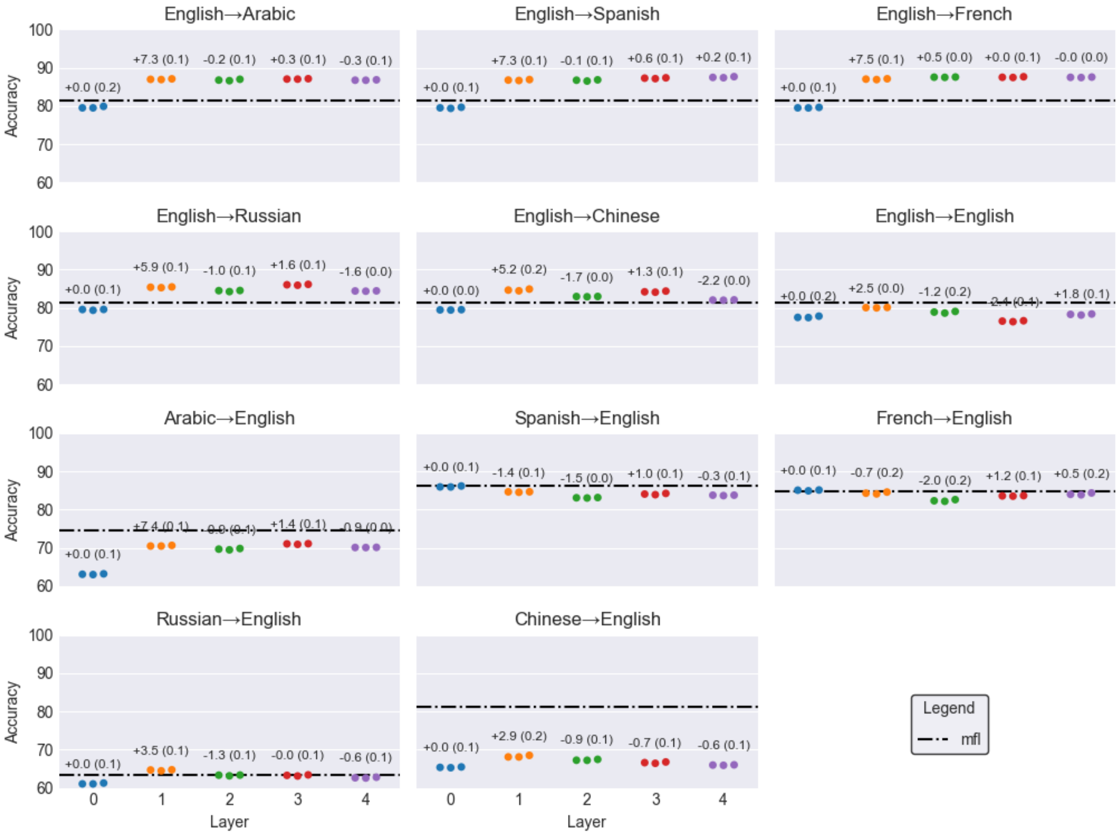


Figure C-1: Full results of predicting morphological tags using encoder representations from different layers of neural machine translation models trained with different target languages (Chapter 4).

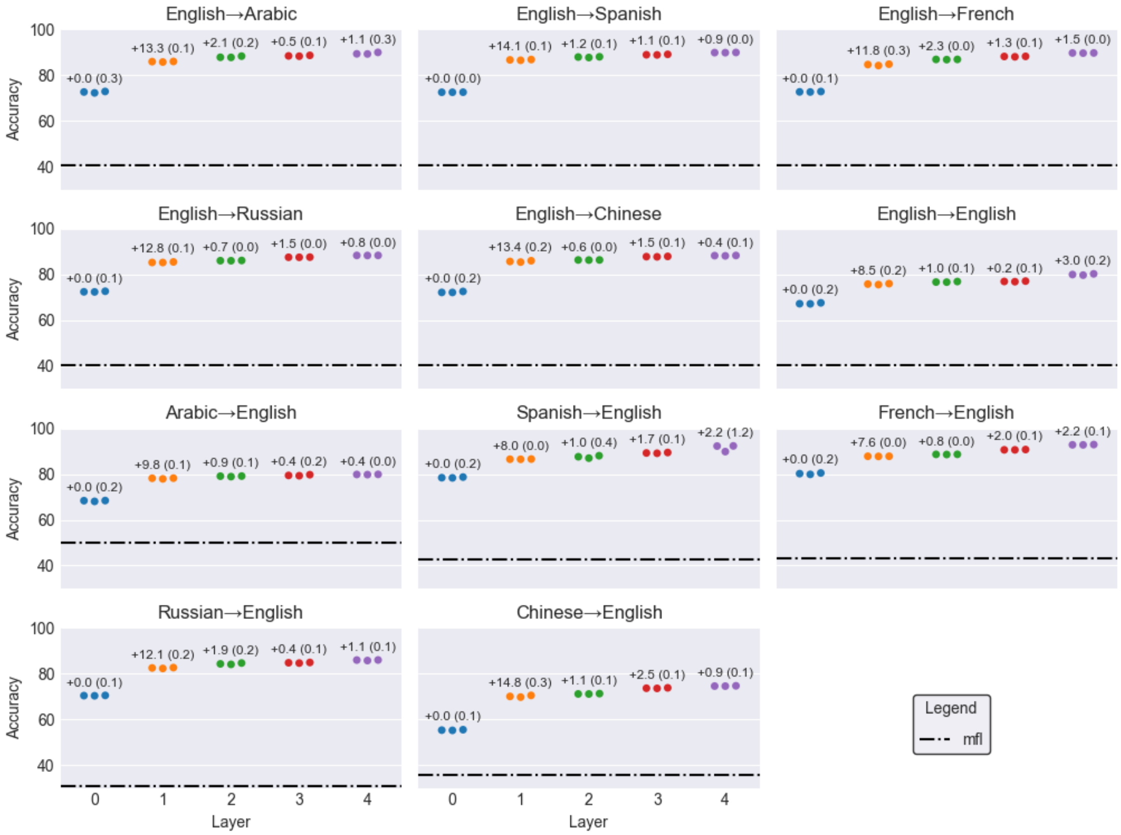


Figure C-2: Full results of predicting syntactic dependencies using encoder representations from different layers of neural machine translation models trained with different target languages (Chapter 4).

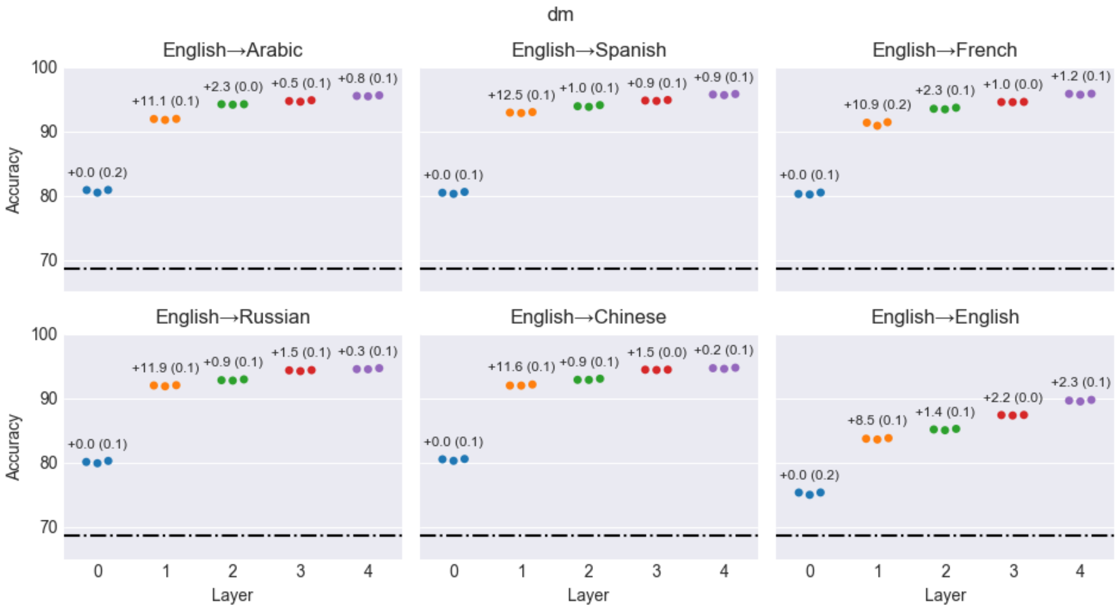


Figure C-3: Full results of predicting semantic dependencies in the DM formalism using encoder representations from different layers of neural machine translation models trained with different target languages (Chapter 4). The dashed line shows the most frequent label baseline.

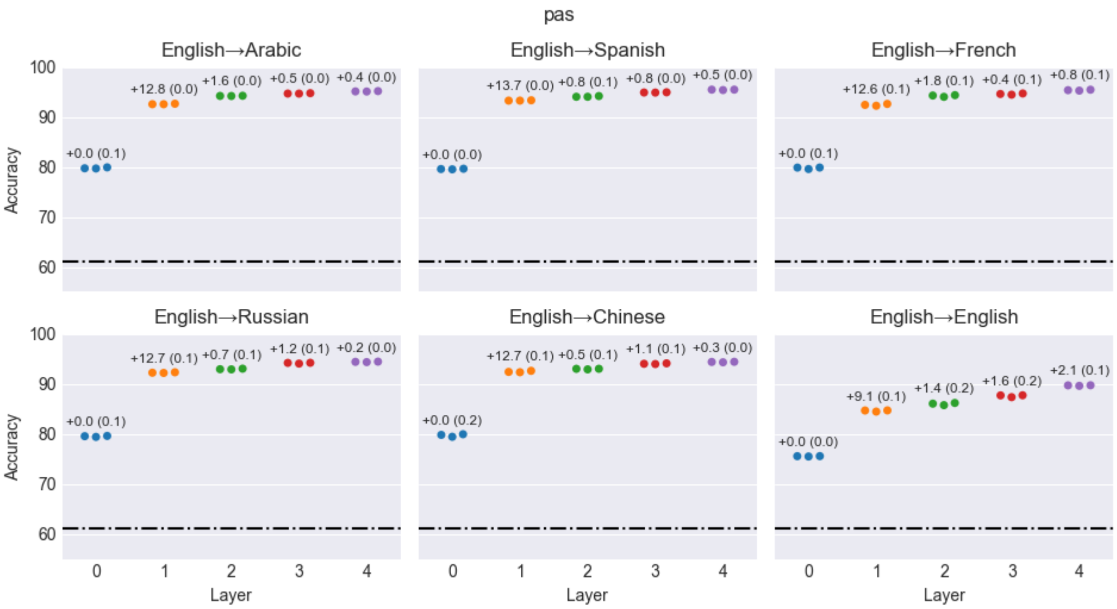


Figure C-4: Full results of predicting semantic dependencies in the PAS formalism using encoder representations from different layers of neural machine translation models trained with different target languages (Chapter 4). The dashed line shows the most frequent label baseline.

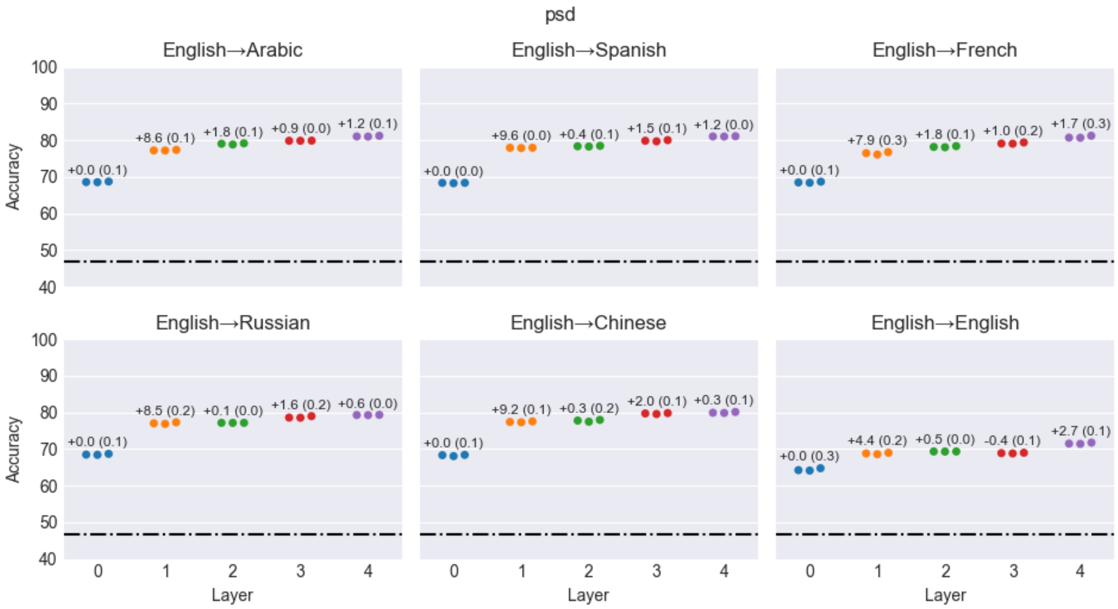


Figure C-5: Full results of predicting semantic dependencies in the PSD formalism using encoder representations from different layers of neural machine translation models trained with different target languages (Chapter 4). The dashed line shows the most frequent label baseline.

C.2 Statistical significance

This section proves a detailed account of statistical significance results for the experiments reported in Chapter 4. There were three independent runs with different random initializations of the classifier for each configuration. A configuration relates to evaluating representations generated from a certain layer of a specific machine translation model, such as layer 1 of the encoder in an English-to-French model. To compare the results between two layers, I choose the two closest runs in terms of accuracy. For each run, I define a binary variable that takes 1 when the prediction is correct, and 0 otherwise. The binary variables corresponding to the two closest runs are compared using the approximate randomization test [265], which has been recommended for computing statistical significance in classification problems in natural language processing.[1]

The statistical significance results for morphology, syntactic dependencies, and semantic dependencies are summarized in Tables C.1a, C.1b, and C.2 respectively.

[1] An implementation is available at `https://www.nlpado.de/~sebastian/software/sigf.shtml`.

English-Arabic

	0	1	2	3	4
0		‡			
1	‡		ns	ns	ns
2		ns		ns	ns
3		ns	ns		ns
4		ns	ns	ns	

English-Spanish

	0	1	2	3	4
0		‡			
1	†		ns	ns	ns
2		‡		ns	*
3		ns	ns		ns
4		ns	ns	ns	

English-French

	0	1	2	3	4
0		‡			
1	ns		ns	ns	ns
2		ns		ns	ns
3		ns	ns		ns
4		ns	ns	ns	

English-Russian

	0	1	2	3	4
0		‡			
1	‡		‡	ns	‡
2		‡		‡	ns
3		ns	ns		‡
4		†	ns	ns	

English-Chinese

	0	1	2	3	4
0		‡			
1	‡		‡	ns	‡
2		ns		‡	ns
3		ns	ns		‡
4		‡	†	ns	

English-English

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		‡	ns
3		‡	‡		‡
4		‡	ns	‡	

(a) Morphology

English-Arabic

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		ns	‡
3		‡	ns		‡
4		‡	†	ns	

English-Spanish

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		ns		‡	‡
3		‡	‡		‡
4		‡	‡	‡	

English-French

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		†		‡	‡
3		‡	‡		‡
4		‡	‡	‡	

English-Russian

	0	1	2	3	4
0		‡			
1	‡		†	‡	‡
2		‡		‡	‡
3		‡	ns		†
4		‡	‡	‡	

English-Chinese

	0	1	2	3	4
0		‡			
1	‡		ns	‡	‡
2		ns		‡	‡
3		‡	‡		ns
4		‡	‡	†	

English-English

	0	1	2	3	4
		‡			
1	‡		†	†	‡
2		†		ns	‡
3		†	ns		‡
4		‡	‡	‡	

(b) Syntax

Table C.1: Statistical significance results for morphological tagging **(a)** and syntactic dependency labeling **(b)** experiments in Chapter 4. In each table with caption A-B, the cells above the main diagonal are for translation direction A→B and those below it are for the direction B→A. ns = $p > 0.05$, * = $p < 0.05$, †= $p < 0.01$, ‡= $p < 0.001$. Comparisons at empty cells are not shown.

English→Arabic/Russian

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		†	‡
3		‡	‡		‡
4		‡	‡	ns	

English→Spanish/Chinese

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		‡	‡
3		‡	‡		‡
4		‡	‡	ns	

English→French/English

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		‡	‡
3		‡	‡		‡
4		‡	‡	‡	

(a) DM scheme

English→Arabic/Russian

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		‡	‡
3		‡	‡		‡
4		‡	‡	ns	

English→Spanish/Chinese

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		‡	‡
3		‡	‡		‡
4		‡	‡	ns	

English→French/English

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		‡		ns	‡
3		‡	‡		‡
4		‡	‡	‡	

(b) PAS scheme

English→Arabic/Russian

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		ns		†	‡
3		‡	‡		‡
4		‡	‡	ns	

English→Spanish/Chinese

	0	1	2	3	4
0		‡			
1	‡		ns	‡	‡
2		ns		‡	‡
3		‡	‡		‡
4		‡	‡	ns	

English→French/English

	0	1	2	3	4
0		‡			
1	‡		‡	‡	‡
2		ns		‡	‡
3		ns	*		‡
4		‡	‡	‡	

(c) PSD scheme

Table C.2: Statistical significance results for semantic dependency labeling experiments in Chapter 4. In each table with caption A→B/C, the cells above the main diagonal are for translation direction A→B and those below the main diagonal are for the direction A→C. ns = $p > 0.05$, * = $p < 0.05$, †= $p < 0.01$, ‡= $p < 0.001$. Comparisons at empty cells are not shown.

C.3 Results by relation type

Figure C-7 shows the improvement in accuracy using representations from layers 2/3/4 compared to layer 1, when predicting different syntactic relations. Figures C-8, C-9, and C-10 show similar numbers for predicting different semantic relations. See the following section for information on the specific relations.

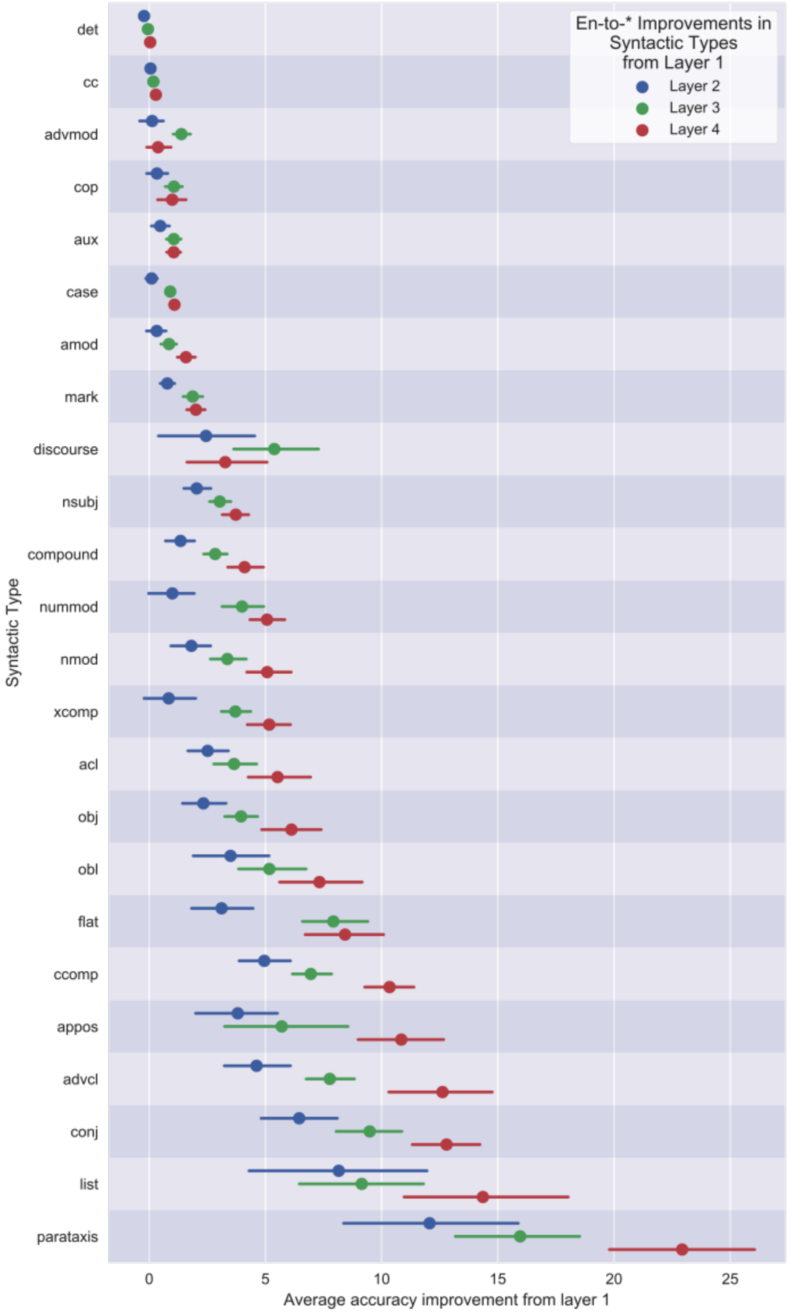


Figure C-6: Accuracy improvement when predicting different syntactic relation types using representations from layers 2/3/4 compared to layer 1, generated by neural machine translation models trained to translate English to other languages.

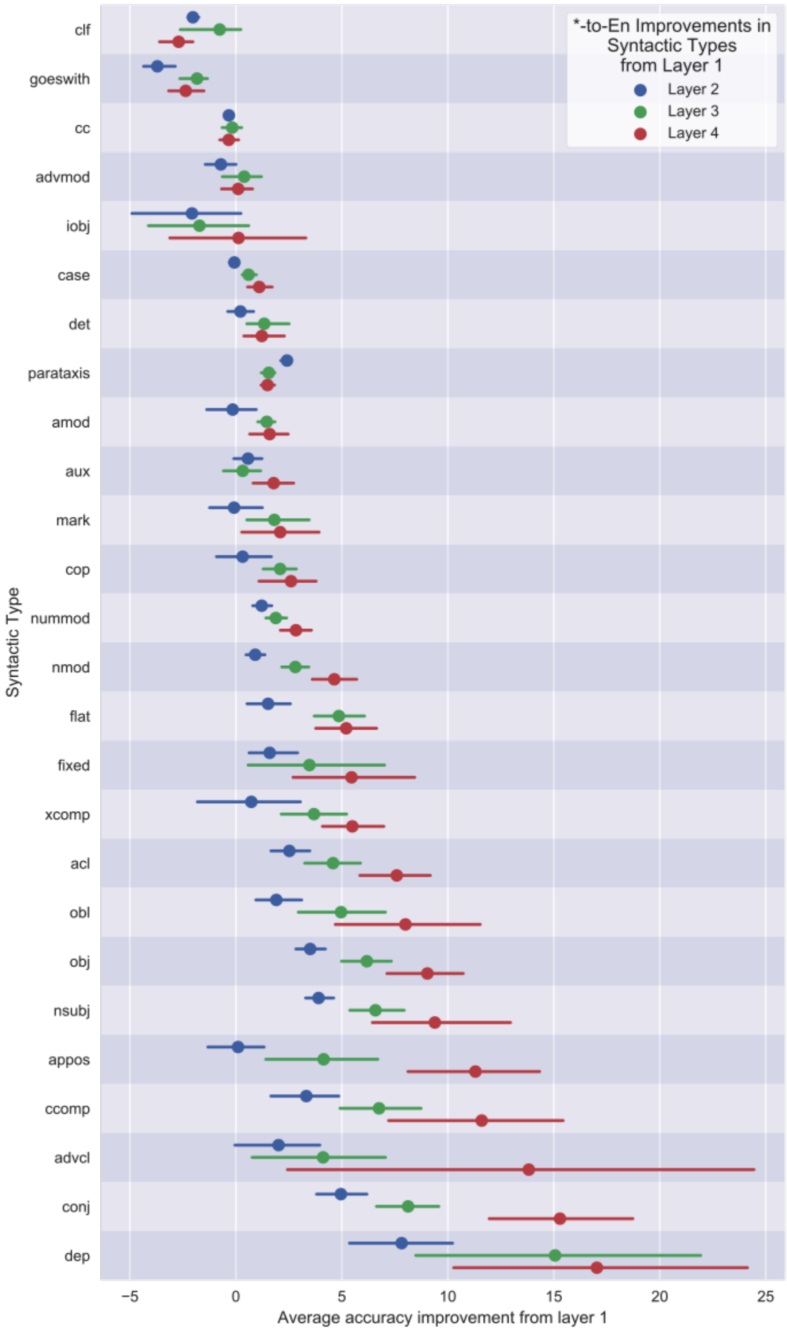


Figure C-7: Accuracy improvement when predicting different syntactic relation types using representations from layers 2/3/4 compared to layer 1, generated by neural machine translation models trained to translate from other languages to English.

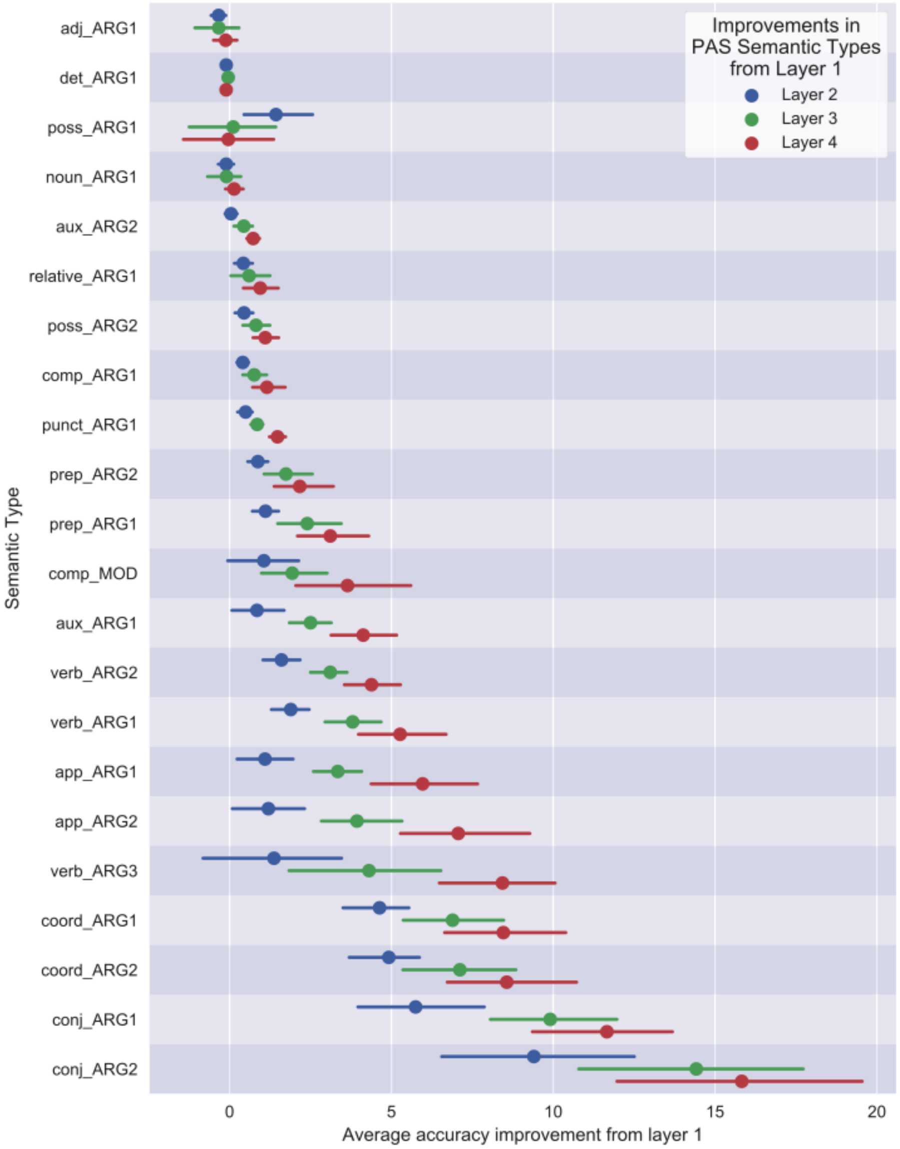


Figure C-8: Accuracy improvement when predicting different English semantic relations (PAS scheme) using representations from layers 2/3/4 compared to layer 1.

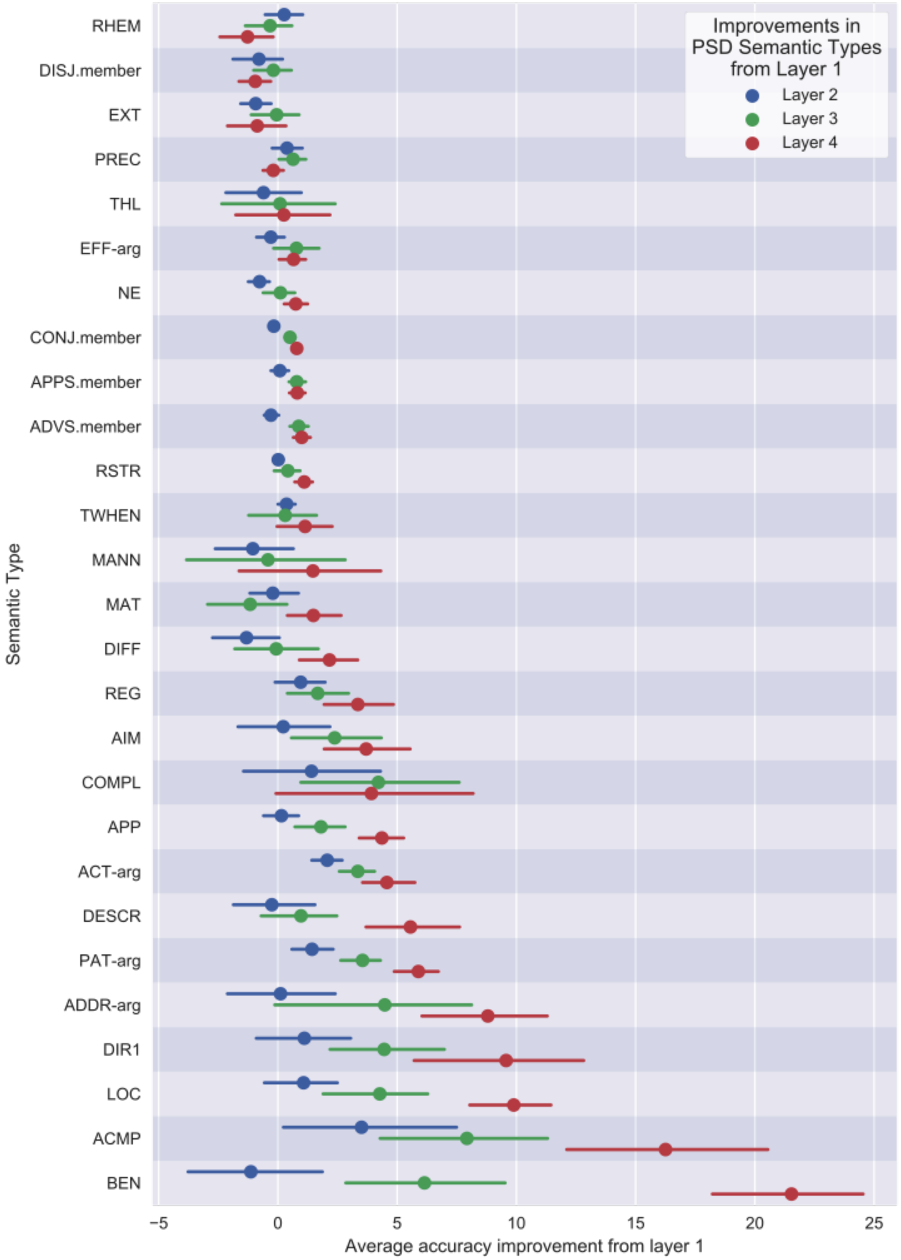


Figure C-9: Accuracy improvement when predicting different English semantic relations (PSD scheme) using representations from layers 2/3/4 compared to layer 1.

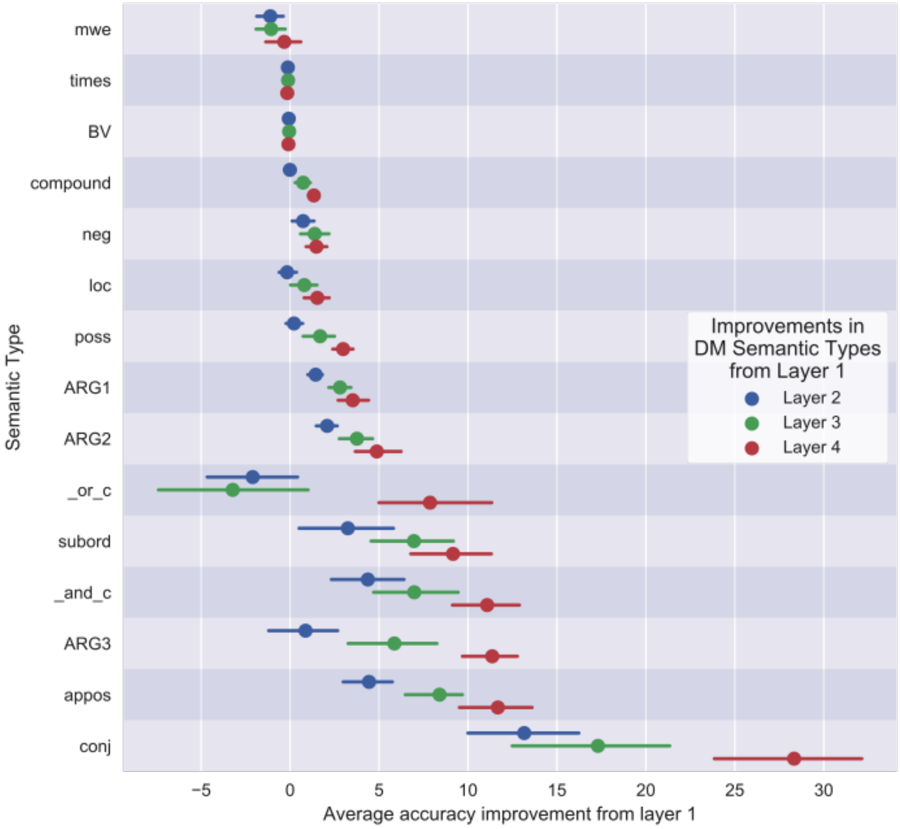


Figure C-10: Accuracy improvement when predicting different English semantic relations (PSD scheme) using representations from layers 2/3/4 compared to layer 1.

C.4 Information on dependency relations

This section lists the syntactic and semantic relations that are mentioned in Chapter 4 and in Figures C-7, C-8, C-9 and C-10. More information is available in the official documentation of the original datasets.

Table C.3 lists the syntactic dependencies from the Universal Dependencies datasets [261]. Consult the online documentation for detailed definitions and examples:
`http://universaldependencies.org`.

For details on the semantic dependency formalisms, see the references on the shared-task website: `http://sdp.delph-in.net/2015/representations.html`. The PAS and DM schemes mainly denote relations by first, second, and third arguments (`ARG1`, `ARG2`, `ARG3`). In the PAS scheme, these are categorized by POS tag. In the DM scheme, there are a few additional, more syntactically oriented relations: multi-word expressions (`mwe`), certain number expressions (`times`), bound variable of a quantifier (`BV`) [162], negation (`neg`), time adverbs (`loc`), possessives (`poss`), the relations between disjuncts (`_or_c`) and conjuncts (`_and_c`, `conj`), subordination (`subord`), and apposition (`appos`). Table C.4 lists the PSD relations that are mentioned in Figure **??**. They are derived from the tectogrammatical layer in the English part of the Prague Czech-English dependency treebank.[2] The manual contains detailed definitions and examples.[3]

[2] `http://ufal.ms.mff.cuni.cz/pcedt2.0/`

[3] See `http://ufal.ms.mff.cuni.cz/pcedt2.0/publications/TR_En.pdf`. The NE relation is not mentioned in the manual, but observing the data shows that it is used for named entity parts like the relation between “South” and “Korea”.

Relation	Description
acl	clausal modifier of noun
advcl	adverbial clause modifier
advmod	adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
case	case marking
cc	coordinating conjunction
ccomp	clausal complement
clf	classifier
compound	compound
conj	conjunct
cop	copula
csubj	clausal subject
dep	unspecified dependency
det	determiner
discourse	discourse element
dislocated	dislocated elements
expl	expletive
fixed	fixed multiword expression
flat	flat multiword expression
goeswith	goes with
iobj	indirect object
list	list
mark	marker
nmod	nominal modifier
nsubj	nominal subject
nummod	numeric modifier
obj	object
obl	oblique nominal
orphan	orphan
parataxis	parataxis
punct	punctuation
reparandum	overridden disfluency
root	root
vocative	vocative
xcomp	open clausal complement

Table C.3: Syntactic dependency relations defined in the Universal Dependencies datasets.

Relation	Description	Relation	Description
ACMP	accompaniment	EFF	functor used for arguments with the cognitive role of the effect/result of the event
ACT	functor for the first argument	EXT	extent
ADDR	functor used for arguments with the cognitive role of the recipient of the event	LOC	where?
ADVS	adversative	MANN	manner proper
AIM	purpose, aim	MAT	adnominal argument referring to the content (material etc.) of something
APP	adjunct referring to the person or thing something or someone belongs to	NE	named entity?
APPS	apposition	PAT	functor for the second argument
BEN	adjunct expressing to whose advantage or disadvantage something happens	PREC	expression linking the clause to the preceding text
COMPL	predicative complement	REG	regard
CONJ	simple conjoining	RHEM	rhematizer
DESCR	nonrestrictive attribute in postposition	RSTR	adnominal adjunct more closely specifying
DIFF	difference	THL	how long? in what time?
DIR1	Where from?	TWHEN	When?
DISJ	disjunctive		

Table C.4: Semantic dependency relations used in the PSD scheme.

Appendix D

ASR Experiments

D.1 Comparing linear and non-linear classifiers

Table D.1 shows a comparison of a linear classifier with two non-linear classifiers, having one and two hidden layers, on frame classification with representations from different layers of DeepSpeech2. The non-linear classifiers perform better at every layer. However, the layer-wise trends are similar, and consistent with the main experiments reported in Chapter 5. Adding a second hidden layer only slightly improves the results.

	Input	cnn1	cnn2	rnn1	rnn2	rnn3	rnn4	rnn5	rnn6	rnn7
Linear	31.57	39.72	14.87	21.12	24.18	25.75	27.12	29.86	24.54	19.18
MLP-1	36.64	45.99	16.86	23.19	27.22	29.39	32.08	33.67	26.96	21.57
MLP-2	38.04	47.61	17.52	23.71	28.02	29.92	32.69	34.02	28.01	21.94

Table D.1: Frame classification accuracy using representations from different layers of DeepSpeech2, as obtained by a linear classifier compared to non-linear multi-layer perceptrons (MLP) with one and two hidden layers. The non-linear classifiers obtain consistently better results than the linear one, but the relative trends (which layers perform better) are similar in both cases.

D.2 Visualizations of frame representations

Figure D-1 shows a 2D projection of centroids of frame representation clusters from different layers of an end-to-end ASR model. See Section 5.6.5 for a description of this visualization. Figure D-2 shows similar visualizations after pruning very impure clusters, ones with a majority label smaller than 10–20% of cluster members.

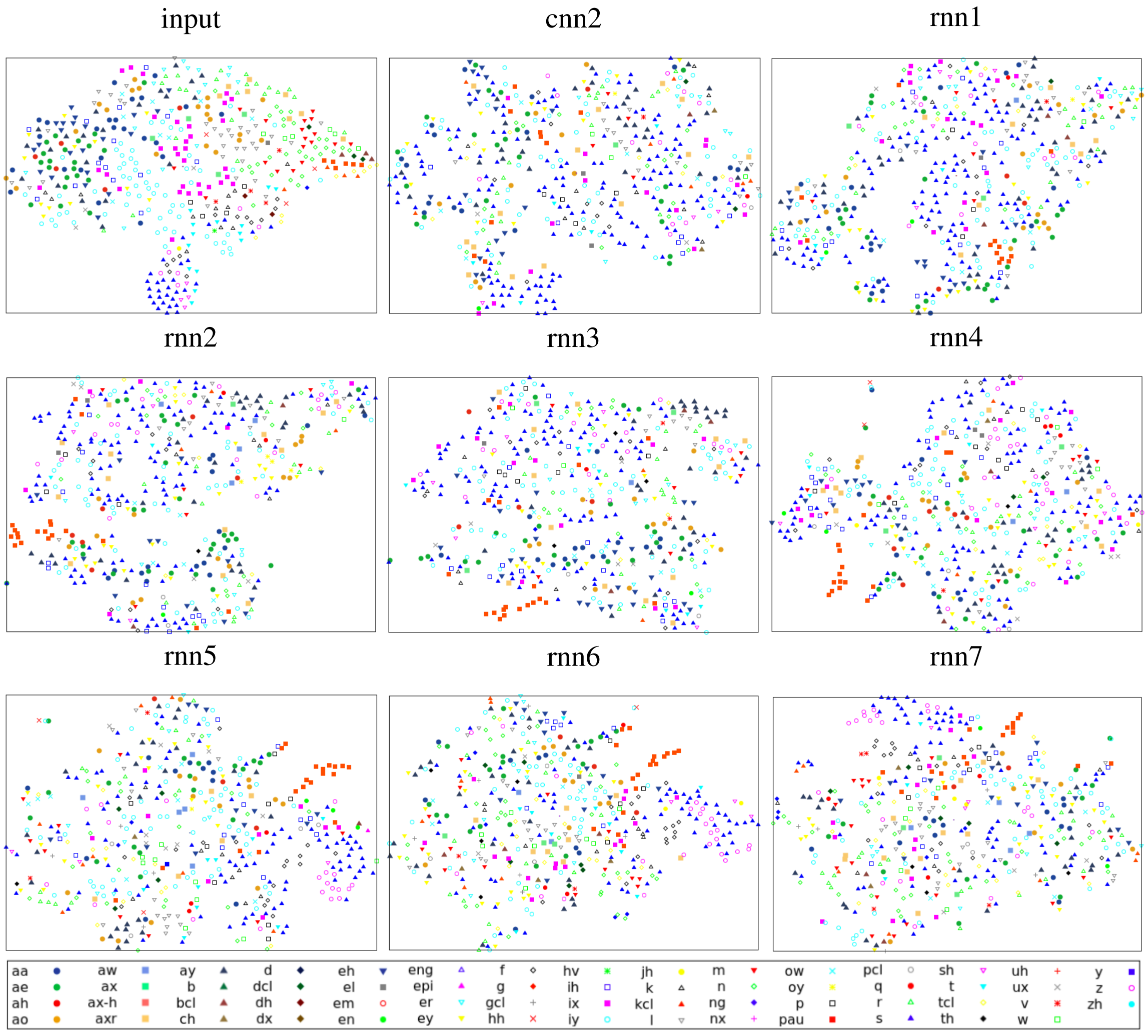


Figure D-1: Centroids of all frame representation clusters using features from different layers of the DeepSpeech2 ASR model (Chapter 5).

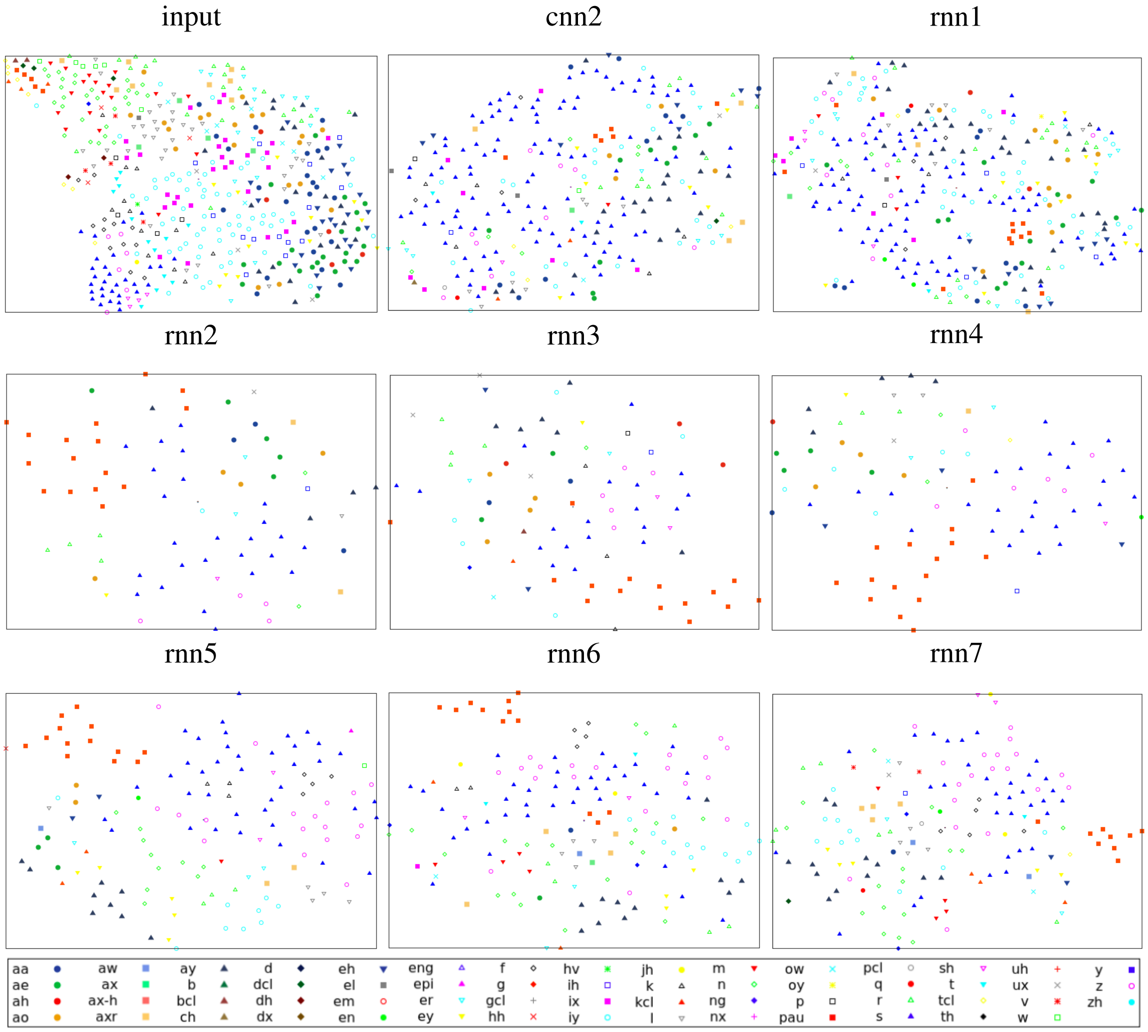


Figure D-2: Centroids of frame representation clusters using features from different layers, showing only clusters where the majority label covers at least 10–20% of the cluster members (Chapter 5).

Bibliography

[1] Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/E17-2039`.

[2] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Analysis of sentence embedding models using prediction tasks in natural language processing. *IBM Journal of Research and Development*, 61(4):3–1, 2017.

[4] Roee Aharoni and Yoav Goldberg. Towards String-To-Tree Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2021. URL `http://www.aclweb.org/anthology/P17-2021`.

[5] Afra Alishahi, Marie Barking, and Grzegorz Chrupała. Encoding of phonology in a recurrent neural model of grounded speech. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 368–378. Association for Computational Linguistics, 2017. doi: 10.18653/v1/K17-1037.

[6] Amjad Almahairi, Cho Kyunghyun, Nizar Habash, and Aaron Courville. First Result on Arabic Neural Machine Translation. *https://arxiv.org/abs/1606.02680*, 2016.

[7] David Alvarez-Melis and Tommi Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1042.

[8] Silvio Amir, Miguel B. Almeida, Bruno Martins, João Filgueiras, and Mario J. Silva. TUGAS: Exploiting unlabelled data for Twitter sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 673–677, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL http://www.aclweb.org/anthology/S14-2120.

[9] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, JingDong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 173–182, 2016.

[10] Leila Arras, Franziska Horn, Grgoire Montavon, Klaus-Robert Mller, and Wojciech Samek. ”What is relevant in a text document?”: An interpretable machine learning approach. *PLOS ONE*, 12(8):1–23, 08 2017. doi: 10.1371/journal.pone.0181142. URL https://doi.org/10.1371/journal.pone.0181142.

[11] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining Recurrent Neural Network Predictions in Sentiment Analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-5221.

[12] Malika Aubakirova and Mohit Bansal. Interpreting Neural Networks to Improve Politeness Comprehension. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2035–2041. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1216. URL http://www.aclweb.org/anthology/D16-1216.

[13] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo. Direct Acoustics-to-Word Models for English Conversational Speech Recognition. *arXiv preprint arXiv:1703.07754*, 2017.

[14] Wilker Aziz, Miguel Rios, and Lucia Specia. Shallow Semantic Trees for SMT. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 316–322, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL `http://dl.acm.org/citation.cfm?id=2132960.2133002`.

[15] Ibrahim Badr, Rabih Zbib, and James Glass. Segmentation for English-to-Arabic Statistical Machine Translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 153–156, Columbus, Ohio, 2008. URL `http://dl.acm.org/citation.cfm?id=1557690.1557732`.

[16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[17] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. *arXiv preprint arXiv:1508.04395*, 2015.

[18] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-End Attention-based Large Vocabulary Speech Recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE, 2016.

[19] Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(2):179–190, February 1983. ISSN 0162-8828. doi: 10.1109/TPAMI.1983.4767370. URL `https://doi.org/10.1109/TPAMI.1983.4767370`.

[20] Mona Baker. *In other words: A coursebook on translation*. Routledge, 2018.

[21] Mohit Bansal, Keving Gimpel, and Karen Livescu. Tailoring Continuous Word Representations for Dependency Parsing. In *Proceedings of ACL-14*. Association for Computational Linguistics, 2014.

[22] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June

2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1023.

[23] Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating Discourse Phenomena in Neural Machine Translation. *arXiv preprint arXiv:1711.00513*, 2017.

[24] Marzieh Bazrafshan and Daniel Gildea. Semantic Roles for String to Tree Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–423, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-2074.

[25] Marzieh Bazrafshan and Daniel Gildea. Comparing Representations of Semantic Roles for String-To-Tree Decoding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1786–1791, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1188.

[26] Yonatan Belinkov and Yonatan Bisk. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations (ICLR)*, April 2018.

[27] Yonatan Belinkov and James Glass. Large-Scale Machine Translation between Arabic and Hebrew: Available Corpora and Initial Results. In *Proceedings of the Workshop on Semitic Machine Translation*, pages 7–12, Austin, Texas, November 2016. Association for Computational Linguistics.

[28] Yonatan Belinkov and James Glass. A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial at Coling)*, Osaka, Japan, December 2016.

[29] Yonatan Belinkov and James Glass. Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems. In *Advances in Neural Information Processing Systems (NIPS)*, December 2017.

[30] Yonatan Belinkov, Tao Lei, Regina Barzilay, and Amir Globerson. Exploring Compositional Architectures and Word Vector Representations for Prepositional Phrase Attachment. *Transactions of the Association for Computational Linguistics*, 2:561–572, 2014. ISSN 2307-387X. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/488.

[31] Yonatan Belinkov, Mitra Mohtarami, Scott Cyphers, and James Glass. VectorSLU: A Continuous Word Vector Approach to Answer Selection in Community Question Answering Systems. *The 9th Workshop on Semantic Evaluation (SemEval-2015)*, 2015.

[32] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, July 2017. Association for Computational Linguistics. URL `https://aclanthology.coli.uni-saarland.de/pdf/P/P17/P17-1080.pdf`.

[33] Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*, November 2017.

[34] Yoshua Bengio, Jrme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *In ICML*, 2009.

[35] Jonathan Berant and Percy Liang. Semantic Parsing via Paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-1133`.

[36] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning Attacks Against Support Vector Machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1467–1474, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1. URL `http://dl.acm.org/citation.cfm?id=3042573.3042761`.

[37] Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Layer-Wise Relevance Propagation for Deep Neural Network Architectures. In *Information Science and Applications (ICISA) 2016*, pages 913–922. Springer, 2016.

[38] Alexandra Birch, Miles Osborne, and Philipp Koehn. CCG Supertags in Factored Statistical Machine Translation. In *Proceedings of the Second Workshop on Statis-*

tical Machine Translation, pages 9–16. Association for Computational Linguistics, 2007. URL `http://www.aclweb.org/anthology/W07-0702`.

[39] Johannes Bjerva, Barbara Plank, and Johan Bos. Semantic Tagging with Deep Residual Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `http://aclweb.org/anthology/C16-1333`.

[40] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012.

[41] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/W17-4717`.

[42] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1075. URL `http://www.aclweb.org/anthology/D15-1075`.

[43] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive Exploration of Neural Machine Translation Architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/D17-1151`.

[44] Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2), 1993. URL `http://www.aclweb.org/anthology/J93-2003`.

[45] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2), 1990. URL `http://www.aclweb.org/anthology/J90-2002`.

[46] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra, and Jenifer C. Lai. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–480, 1992. URL http://www.aclweb.org/anthology/J92-4003.

[47] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Michael Weigelt. Natural Language Multitasking: Analyzing and Improving Syntactic Saliency of Hidden Representations. *arXiv preprint arXiv:1801.06024*, 2018.

[48] Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 2.0, 2004. LDC Catalog No. LDC2004L02.

[49] Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108(1): 159–170, 2017.

[50] Franck Burlot and François Yvon. Evaluating the morphological competence of Machine Translation Systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-4705.

[51] Nicholas Carlini and David Wagner. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. *arXiv preprint arXiv:1801.01944*, 2018.

[52] Marine Carpuat and Dekai Wu. Improving Statistical Machine Translation Using Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007. URL http://www.aclweb.org/anthology/D07-1007.

[53] Xavier Carreras and Michael Collins. Non-Projective Parsing for Statistical Machine Translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 200–209. Association for Computational Linguistics, 2009. URL http://www.aclweb.org/anthology/D09-1021.

[54] Asuncion Castano and Francisco Casacuberta. A connectionist approach to machine translation. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[55] Mauro Cettolo. An Arabic-Hebrew parallel corpus of TED talks. In *Proceedings of the AMTA Workshop on Semitic Machine Translation (SeMaT)*, Austin, US-TX, November 2016.

[56] Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT^3: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May 2012.

[57] Rahma Chaabouni, Ewan Dunbar, Neil Zeghidour, and Emmanuel Dupoux. Learning weakly supervised multimodal phoneme embeddings. In *Interspeech 2017*, 2017.

[58] Seng Yee Chan, Tou Hwee Ng, and David Chiang. Word Sense Disambiguation Improves Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40. Association for Computational Linguistics, 2007. URL `http://aclweb.org/anthology/P07-1005`.

[59] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, Attend and Spell. *arXiv preprint arXiv:1508.01211*, 2015.

[60] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf`.

[61] Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. Optimizing Chinese Word Segmentation for Machine Translation Performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 224–232, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1. URL `http://dl.acm.org/citation.cfm?id=1626394.1626430`.

[62] Danqi Chen and Christopher D Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[63] Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. Neural Machine Translation with Source Dependency Representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2846–2852. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/D17-1304`.

[64] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung-yi Lee. Towards Unsupervised Automatic Speech Recognition Trained by Unaligned Speech and Text only. *arXiv preprint arXiv:1803.10952*, 2018.

[65] David Chiang. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, 2005. doi: 10.3115/1219840.1219873. URL `https://doi.org/10.3115/1219840.1219873`.

[66] David Chiang. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2):201–228, 2007.

[67] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art Speech Recognition With Sequence-to-Sequence Models. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[68] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4012. URL `http://www.aclweb.org/anthology/W14-4012`.

[69] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*, 2014.

[70] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. *arXiv preprint arXiv:1412.1602*, 2014.

[71] Wu Chou and Biing-Hwang Juang. Minimum Classification Error (MCE) Approach in Pattern Recognition. In *Pattern Recognition in Speech and Language Processing*, pages 12–58. CRC Press, 2003.

[72] Lonnie Chrisman. Learning Recursive Distributed Representations for Holistic Computation. *Connection Science*, 3(4):345–366, 1991. doi: 10.1080/09540099108946592. URL https://doi.org/10.1080/09540099108946592.

[73] Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 613–622. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1057.

[74] Grzegorz Chrupała, Lieke Gelderloos, Ákos Kádár, and Afra Alishahi. On the difficulty of a distributional semantics of spoken language. *arXiv preprint arXiv:1803.08869*, 2018.

[75] Yu-An Chung and James Glass. Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech. *arXiv preprint arXiv:1803.08976*, 2018.

[76] Moustapha M. Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling Deep Structured Visual and Speech Recognition Models with Adversarial Examples. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6977–6987. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7273-houdini-fooling-deep-structured-visual-and-speech-recognition-models-with-adversarial-examples.pdf.

[77] Ronan Collobert and Jason Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. URL http://doi.acm.org/10.1145/1390156.1390177.

[78] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from

Scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078186.

[79] Costanza Conforti, Matthias Huck, and Alexander Fraser. Neural Morphological Tagging of Lemma Sequences for Machine Translation. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas (Volume 1: Research Track*, pages 39–53, March 2018.

[80] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1070.

[81] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2018.

[82] Marta R. Costa-jussà and José A. R. Fonollosa. Character-based Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-2058.

[83] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*, November 2017.

[84] Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating Compositionality in Sentence Embeddings. *arXiv preprint arXiv:1802.04302*, 2018.

[85] K. H. Davis, R. Biddulph, and S. Balashek. Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[86] Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. Fast and Accurate Preordering for SMT using Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 1012–1017. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1105. URL `http://www.aclweb.org/anthology/N15-1105`.

[87] Claudio Delli Bovi, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–600. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2094. URL `http://www.aclweb.org/anthology/P17-2094`.

[88] P. Denes. The Design and Operation of the Mechanical Speech Recognizer at University College London. *Journal of the British Institution of Radio Engineers*, 19 (4):219–229, 1959.

[89] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380. Association for Computational Linguistics, 2014. doi: 10.3115/v1/P14-1129. URL `http://www.aclweb.org/anthology/P14-1129`.

[90] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and Understanding Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1106. URL `http://www.aclweb.org/anthology/P17-1106`.

[91] Yuan Ding and Martha Palmer. Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548. Association for Computational Linguistics, 2005. URL `http://www.aclweb.org/anthology/P05-1067`.

[92] Cicero dos Santos and Maira Gatti. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/C14-1008`.

[93] Cicero dos Santos, Bing Xiang, and Bowen Zhou. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1061`.

[94] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. In *arXiv preprint arXiv:1702.08608*, 2017.

[95] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Shieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of AI Under the Law: The Role of Explanation. *Berkman Center Publication Forthcoming*, 2017.

[96] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. Using Latent Semantic Analysis To Improve Access To Textual Information. In *SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*, pages 281–285. ACM, 1988.

[97] Nadir Durrani, Hassan Sajjad, Alexander Fraser, and Helmut Schmid. Hindi-to-Urdu Machine Translation through Transliteration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 465–474, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1048`.

[98] Nadir Durrani, Philipp Koehn, Helmut Schmid, and Alexander Fraser. Investigating the Usefulness of Generalized Word Representations in SMT. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 421–432, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/C14-1041`.

[99] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-Box Adversarial Examples for NLP. *arXiv preprint arXiv:1712.06751*, 2017.

[100] Jason Eisner. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, 2003. URL `http://www.aclweb.org/anthology/P03-2041`.

[101] Jeffrey L. Elman. Representation and Structure in Connectionist Models. Technical report, University of California, San Diego, Center for Research in Language, 1989.

[102] Jeffrey L. Elman. Finding Structure in Time. *Cognitive science*, 14(2):179–211, 1990.

[103] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3):195–225, 1991.

[104] Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1078. URL `http://www.aclweb.org/anthology/P16-1078`.

[105] Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to Parse and Translate Improves Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2012. URL `http://www.aclweb.org/anthology/P17-2012`.

[106] Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-2524. URL `http://www.aclweb.org/anthology/W16-2524`.

[107] Florian Eyben, Martin Wöllmer, Björn Schuller, and Alex Graves. From Speech to Letters - Using a Novel Neural Network Architecture for Grapheme Based ASR. In *2009 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 376–380, Nov 2009. doi: 10.1109/ASRU.2009.5373257.

[108] Mark Fishel and Harri Kirik. Linguistically Motivated Unsupervised Segmentation for Machine Translation. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, 2010.

[109] Mikel L. Forcada and Ramón P. Ñeco. Recursive Hetero-associative Memories for Translation. In *Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Biological and Artificial Computation: From Neuroscience to Technology*, IWANN '97, pages 453–462, London, UK, UK,

1997. Springer-Verlag. ISBN 3-540-63047-3. URL `http://dl.acm.org/citation.cfm?id=646367.690318`.

[110] Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. Modeling Inflection and Word-Formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674. Association for Computational Linguistics, 2012. URL `http://www.aclweb.org/anthology/E12-1068`.

[111] Dennis Butler Fry. Theoretical Aspects of Mechanical Speech Recognition. *Journal of the British Institution of Radio Engineers*, 19(4):211–218, 1959.

[112] Sadaoki Furui. History and Development of Speech Recognition. In Fang Chen, editor, *Speech Technology: Theory and Applications*, pages 1–18. Springer US, Boston, MA, 2010. ISBN 978-0-387-73819-2. doi: 10.1007/978-0-387-73819-2_1. URL `https://doi.org/10.1007/978-0-387-73819-2_1`.

[113] Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. A Compositional and Interpretable Semantic Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1004. URL `http://www.aclweb.org/anthology/N15-1004`.

[114] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 2004. URL `http://www.aclweb.org/anthology/N04-1035`.

[115] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968. Association for Computational Linguistics, 2006. URL `http://www.aclweb.org/anthology/P06-1121`.

[116] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. Learning Generic Sentence Representations Using Convolutional Neural

Networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2390–2400. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1254.

[117] J. Ganesh, Manish Gupta, and Vasudeva Varma. Interpretation of Semantic Tweet Representations. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 95–102, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4993-2. doi: 10.1145/3110025.3110083. URL http://doi.acm.org/10.1145/3110025.3110083.

[118] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. *arXiv preprint arXiv:1801.04354*, 2018.

[119] Qin Gao and Stephan Vogel. Utilizing Target-Side Semantic Role Labels to Assist Hierarchical Phrase-based Machine Translation. In *Proceedings of Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 107–115. Association for Computational Linguistics, 2011. URL http://aclweb.org/anthology/W11-1012.

[120] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, and David S. Pallett. TIMIT Acoustic-Phonetic Continuous Speech Corpus , 1993. LDC Catalog No. LDC93S1.

[121] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. A Convolutional Encoder Model for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1012. URL http://www.aclweb.org/anthology/P17-1012.

[122] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/gehring17a.html.

[123] Lieke Gelderloos and Grzegorz Chrupała. From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning.

In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1309–1319, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `http://aclweb.org/anthology/C16-1124`.

[124] James R Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech & Language*, 17(2):137 – 152, 2003. ISSN 0885-2308. doi: https://doi.org/10.1016/S0885-2308(03)00006-8. URL `http://www.sciencedirect.com/science/article/pii/S0885230803000068`. New Computational Paradigms for Acoustic Modeling in Speech Recognition.

[125] Ben Gold, Nelson Morgan, and Dan Ellis. Brief History of Automatic Speech Recognition. In *Speech and Audio Signal Processing: Processing and Perception of Speech and Music,*, chapter 5. John Wiley & Sons, 2 edition, 2011.

[126] David Golub and Xiaodong He. Character-Level Question Answering with Attention. *arXiv preprint arXiv:1604.00727*, 2016.

[127] Yuan Gong and Christian Poellabauer. An Overview of Vulnerabilities of Voice Controlled Systems. *arXiv preprint arXiv:1803.09156*, 2018.

[128] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[129] Alex Graves. Generating Sequences with Recurrent Neural Networks. *arXiv preprint arXiv:1308.0850*, 2013.

[130] Alex Graves and Navdeep Jaitly. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772. JMLR Workshop and Conference Proceedings, 2014.

[131] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[132] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

[133] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of ICASSP*, 2013.

[134] Spence Green and John DeNero. A Class-based Agreement Model for Generating Accurately Inflected Translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 146–155, 2012.

[135] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1l8BtlCb`.

[136] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

[137] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2018.

[138] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1002. URL `http://www.aclweb.org/anthology/D15-1002`.

[139] Nizar Habash and Fatiha Sadat. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'06)*, New York, NY, USA, 2006.

[140] Nizar Y. Habash. Introduction to Arabic Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187, 2010. doi: 10.2200/S00277ED1V01Y201008HLT010. URL `https://doi.org/10.2200/S00277ED1V01Y201008HLT010`.

[141] David Harwath and James Glass. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 237–244, Dec 2015. doi: 10.1109/ASRU.2015.7404800.

[142] David Harwath and James Glass. Learning Word-Like Units from Joint Audio-Visual Analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–517. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1047.

[143] David Harwath, Antonio Torralba, and James Glass. Unsupervised Learning of Spoken Language with Visual Context. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1858–1866. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6186-unsupervised-learning-of-spoken-language-with-visual-context.pdf`.

[144] Kazuma Hashimoto and Yoshimasa Tsuruoka. Neural Machine Translation with Source-Side Latent Graph Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 125–135. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/D17-1012`.

[145] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[146] Georg Heigold, Günter Neumann, and Josef van Genabith. How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse? In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas (Volume 1: Research Track*, pages 68–79, March 2018.

[147] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N16-1162`.

[148] Felix Hill, Kyunghyun Cho, Sébastien Jean, and Yoshua Bengio. The representational geometry of word meanings acquired by neural machine translation

models. *Machine Translation*, 31(1):3–18, Jun 2017. ISSN 1573-0573. doi: 10.1007/s10590-017-9194-2. URL `https://doi.org/10.1007/s10590-017-9194-2`.

[149] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[150] Liang Huang, Kevin Knight, and Aravind Joshi. A Syntax-Directed Translator with Extended Domain of Locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8. Association for Computational Linguistics, 2006. URL `http://www.aclweb.org/anthology/W06-3601`.

[151] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. ISBN 0130226165.

[152] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. Language Modeling. In *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, chapter 11. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. ISBN 0130226165.

[153] Matthias Huck, Simon Riess, and Alexander Fraser. Target-side Word Segmentation Strategies for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/W17-4706`.

[154] Matthias Huck, Aleš Tamchyna, Ondřej Bojar, and Alexander Fraser. Producing Unseen Morphological Variants in Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 369–375. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/E17-2059`.

[155] Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *arXiv preprint arXiv:1711.10203*, 2017.

[156] William John Hutchins. *Machine Translation: Past, Present, Future*. John Wiley & Sons, Inc., New York, NY, USA, 1986. ISBN 0-470-20313-7.

[157] William John Hutchins. *Early years in machine translation: Memoirs and biographies of pioneers*, volume 97. John Benjamins Publishing, 2000.

[158] William John Hutchins and Harold L Somers. *An Introduction to Machine Translation*, volume 362. Academic Press London, 1992.

[159] Nenad Koncar Imperial, Nenad Koncar, and Dr. Gregory Guthrie. A Natural Language Translation Neural Network. In *In Proceedings of the International Conference on New Methods in Language Processing (NeMLaP*, pages 71–77, 1994.

[160] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning (ICML)*, volume 37, pages 448–456, 2015.

[161] Pierre Isabelle, Colin Cherry, and George Foster. A Challenge Set Approach to Evaluating Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/D17-1263`.

[162] Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. Who Did What to Whom? A Contrastive Study of Syntacto-Semantic Dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11. Association for Computational Linguistics, 2012. URL `http://www.aclweb.org/anthology/W12-3602`.

[163] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A Neural Network for Factoid Question Answering over Paragraphs. In *Empirical Methods in Natural Language Processing*, 2014.

[164] Robin Jia and Percy Liang. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2011–2021, Copenhagen, Denmark, September 2017.

[165] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv preprint arXiv:1611.04558*, 2016.

[166] Bevan Jones, Jacob Andreas, Daniel Bauer, Moritz Karl Hermann, and Kevin Knight. Semantics-Based Machine Translation with Hyperedge Replacement Grammars. In *Proceedings of COLING 2012*, pages 1359–1376. The COLING 2012 Organizing Committee, 2012. URL http://aclweb.org/anthology/C12-1083.

[167] Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[168] Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4): 761–780, 2017.

[169] Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. Extractive Summarization using Continuous Vector Space Models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1504.

[170] Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics, 2013. URL http://www.aclweb.org/anthology/D13-1176.

[171] Herman Kamper, Shane Settle, Gregory Shakhnarovich, and Karen Livescu. Visually Grounded Learning of Keyword Prediction from Untranscribed Speech. In *Proc. Interspeech 2017*, pages 3677–3681, 2017. doi: 10.21437/Interspeech.2017-502. URL http://dx.doi.org/10.21437/Interspeech.2017-502.

[172] Shin Kanouchi, Katsuhito Sudoh, and Mamoru Komachi. Neural Reordering Model Considering Phrase Translation and Word Alignment for Phrase-based Translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 94–103. The COLING 2016 Organizing Committee, 2016. URL http://www.aclweb.org/anthology/W16-4607.

[173] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and Understanding Recurrent Networks. *arXiv preprint arXiv:1506.02078*, 2015.

[174] Sameer Khurana, Maryam Najafian, Ahmed Ali, Tuka Al Hanai, Yonatan Belinkov, and James Glass. QMDIS: QCRI-MIT Advanced Dialect Identification System. In *Proceedings of Interspeech*, Stockholm, August 2017.

[175] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1181`.

[176] Yoon Kim. Seq2seq-attn. `https://github.com/harvardnlp/seq2seq-attn`, 2016.

[177] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2741–2749. AAAI Press, 2016. URL `http://dl.acm.org/citation.cfm?id=3016100.3016285`.

[178] Margaret King and Kirsten Falkedal. Using Test Suites in Evaluation of Machine Translation Systems. In *COLNG 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*, 1990. URL `http://www.aclweb.org/anthology/C90-2037`.

[179] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[180] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[181] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[182] Philipp Koehn. Neural Machine Translation. *arXiv preprint arXiv:1709.07809*, 2017.

[183] Philipp Koehn and Hieu Hoang. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D07-1091`.

[184] Philipp Koehn and Kevin Knight. Empirical Methods for Compound Splitting. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–194, 2003. URL `http://www.aclweb.org/anthology/E03-1076`.

[185] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003. URL `http://www.aclweb.org/anthology/N03-1017`.

[186] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[187] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning (ICML)*, 2017.

[188] Arne Köhn. What's in an Embedding? Analyzing Word Embeddings through Multilingual Evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1246`.

[189] Mamoru Komachi, Yuji Matsumoto, and Masaaki Nagata. Phrase Reordering for Statistical Machine Translation Based on Predicate-Argument Structure. In *International Workshop on Spoken Language Translation (IWSLT)*, pages 77–82, 2006.

[190] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. Fooling End-to-end Speaker Verification by Adversarial Examples. *arXiv preprint arXiv:1801.03339*, 2018.

[191] Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. What Do Recurrent Neural Network Grammars Learn About Syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/E17-1117`.

[192] Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv preprint arXiv:1711.00350*, 2018.

[193] César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio. Batch Normalized Recurrent Neural Networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2657–2661. IEEE, 2016.

[194] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association of Computational Linguistics*, 5:365–378, 2017. URL `http://aclweb.org/anthology/Q17-1026`.

[195] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1641–1648, 1989. ISSN 0096-3518. doi: 10.1109/29.46546.

[196] Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Herve Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. TSNLP - Test Suites for Natural Language Processing. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996. URL `http://www.aclweb.org/anthology/C96-2120`.

[197] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1011. URL `http://www.aclweb.org/anthology/D16-1011`.

[198] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf`.

[199] Jiwei Li and Eduard Hovy. A Model of Coherence Based on Distributed Sentence Representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*, 2014. URL `http://emnlp2014.org/papers/pdf/EMNLP2014218.pdf`.

[200] Jiwei Li, Rumeng Li, and Eduard Hovy. Recursive Deep Models for Discourse Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 2061–2069, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1220`.

[201] Jiwei Li, Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1107. URL `http://www.aclweb.org/anthology/P15-1107`.

[202] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. When Are Tree Structures Necessary for Deep Learning of Representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1278`.

[203] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1082. URL `http://www.aclweb.org/anthology/N16-1082`.

[204] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding Neural Networks through Representation Erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[205] Junhui Li, Philip Resnik, and Hal Daumé III. Modeling Syntactic and Semantic Structures in Hierarchical Phrase-based Translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 540–549. Association for Computational Linguistics, 2013. URL `http://www.aclweb.org/anthology/N13-1060`.

[206] Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. A Neural Reordering Model for Phrase-based Translation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1897–1907. Dublin City University and Association for Computational Linguistics, 2014. URL `http://www.aclweb.org/anthology/C14-1179`.

[207] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep Text Classification Can be Fooled. *arXiv preprint arXiv:1704.08006*, 2017.

[208] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1176`.

[209] Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. Character-based Neural Machine Translation. *arXiv preprint arXiv:1511.04586*, 2016.

[210] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

[211] Richard P Lippmann. Neural Network Classifiers for Speech Recognition. *Lincoln Laboratory Journal*, 1:107–124, 1988.

[212] Zachary C Lipton. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2016.

[213] Ding Liu and Daniel Gildea. Improved Tree-to-String Transducer for Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 62–69. Association for Computational Linguistics, 2008. URL `http://www.aclweb.org/anthology/W08-0308`.

[214] Ding Liu and Daniel Gildea. Semantic Role Features for Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 716–724. Coling 2010 Organizing Committee, 2010. URL `http://aclweb.org/anthology/C10-1081`.

[215] Frederick Liu, Han Lu, and Graham Neubig. Handling Homographs in Neural Machine Translation. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2018.

[216] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference*

on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 609–616. Association for Computational Linguistics, 2006. URL `http://www.aclweb.org/anthology/P06-1077`.

[217] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. In *International Conference on Learning Representations (ICLR)*, 2017.

[218] Shixiang Lu, Zhenbiao Chen, and Bo Xu. Learning New Semi-Supervised Deep Auto-encoder Features for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 122–132. Association for Computational Linguistics, 2014. doi: 10.3115/v1/P14-1012. URL `http://www.aclweb.org/anthology/P14-1012`.

[219] Minh-Thang Luong and D. Christopher Manning. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1100. URL `http://aclweb.org/anthology/P16-1100`.

[220] Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. A Hybrid Morpheme-Word Representation for Machine Translation of Morphologically Rich Languages. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 148–157. Association for Computational Linguistics, 2010. URL `http://aclweb.org/anthology/D10-1015`.

[221] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1166`.

[222] Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Ng. Lexicon-Free Conversational Speech Recognition with Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 345–354, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N15-1038`.

[223] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[224] Shozo Makino, Takeshi Kawabata, and Ken'iti Kido. Recognition of consonant based on the perceptron model. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83.*, volume 8, pages 738–741. IEEE, 1983.

[225] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=972470.972475`.

[226] D. Marr and T. Poggio. From understanding computation to understanding neural circuitry. Technical report, Cambridge, MA, USA, 1976.

[227] James H Martin and Daniel Jurafsky. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson/Prentice Hall, 2009.

[228] Rebecca Marvin and Philipp Koehn. Exploring Word Sense Disambiguation Abilities of Neural Machine Translation Systems. In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas (Volume 1: Research Track*, pages 125–131, March 2018.

[229] Ryan McDonald and Joakim Nivre. Analyzing and Integrating Dependency Parsers. *Computational Linguistics*, 37(1), 2011. doi: 10.1162/coli_a_00039. URL `http://www.aclweb.org/anthology/J11-1007`.

[230] Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 216–220. Association for Computational Linguistics, 2006. URL `http://www.aclweb.org/anthology/W06-2932`.

[231] Ian J. Mclean. Example-based machine translation using connectionist matching. In *Proceedings of the Fourth International Conference onTheoretical and Methodological Issues in Machine Translation of Natural Languages*, pages 35–43, 1992.

[232] Shike Mei and Xiaojin Zhu. Using Machine Teaching to Identify Optimal Training-set Attacks on Machine Learners. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2871–2877. AAAI Press,

2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2886521.2886721.

[233] Igor Aleksandrovič Mel'čuk. *Dependency Syntax: Theory and Practice*. SUNY press, 1988.

[234] Yajie Miao, Mohammad Gowayyed, and Florian Metze. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 167–174. IEEE, 2015.

[235] Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. Deep architectures for Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-4710.

[236] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent Neural Network Based Language Model. In *Eleventh Annual Conference of the International Speech Communication Association (Interspeech)*, 2010.

[237] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*, 2013.

[238] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[239] Tim Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences. *arXiv preprint arXiv:1706.07269*, 2017.

[240] Einat Minkov, Kristina Toutanova, and Hisami Suzuki. Generating Complex Morphology for Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 128–135. Association for Computational Linguistics, 2007. URL http://www.aclweb.org/anthology/P07-1017.

[241] Andriy Mnih and Geoffrey Hinton. A scalable hierarchical distributed language model. In *Neural Information Processing Systems (NIPS)*, 2008.

[242] Hans Moen, Erwin Marsi, Filip Ginter, Laura-Maria Murtola, Tapio Salakoski, and Sanna Salanterä. Care Episode Retrieval. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 116–124, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W14-1118`.

[243] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. Understanding how deep belief networks perform acoustic modelling. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4273–4276. IEEE, 2012.

[244] Michael Mohler, Bryan Rink, David Bracewell, and Marc Tomlinson. A Novel Distributional Approach to Multilingual Conceptual Metaphor Recognition. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1752–1763, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/C14-1165`.

[245] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[246] Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Scott Cyphers, James Glass, and Kfir Bar. SLS: Neural-based Approaches to Answer Selection and Question Retrieval in Community Question Answering Systems. *The 10th Workshop on Semantic Evaluation (SemEval-2016)*, 2016.

[247] Grgoire Montavon, Wojciech Samek, and Klaus-Robert Mller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1 – 15, 2018. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2017.10.011. URL `http://www.sciencedirect.com/science/article/pii/S1051200417302385`.

[248] W. James Murdoch, Peter J. Liu, and Bin Yu. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rkRwGg-0Z`.

[249] Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. In *Proceedings of COLING 2012*, pages 1933–1950. The COLING 2012 Organizing Committee, 2012. URL `http://www.aclweb.org/anthology/C12-1118`.

[250] Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Predicting Target Language CCG Supertags Improves Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/W17-4707`.

[251] Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. Exploring How Deep Neural Networks Form Phonemic Categories. In *Interspeech 2015*, 2015.

[252] Tasha Nagamine, Michael L. Seltzer, and Nima Mesgarani. On the Role of Non-linear Transformations in Deep Neural Network Acoustic Models. In *Interspeech 2016*, pages 803–807, 2016. doi: 10.21437/Interspeech.2016-1406.

[253] Preslav Nakov and Jörg Tiedemann. Combining Word-Level and Character-Level Models for Machine Translation Between Closely-Related Languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '12, pages 301–305, Jeju, Korea, 2012. URL `http://aclweb.org/anthology/P12-2059`.

[254] Sean Naren. deepspeech.torch. `https://github.com/SeanNaren/deepspeech.torch`, 2016.

[255] N. Narodytska and S. Kasiviswanathan. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318, July 2017. doi: 10.1109/CVPRW.2017.172.

[256] Graham Neubig. Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96. Association for Computational Linguistics, 2013. URL `http://www.aclweb.org/anthology/P13-4016`.

[257] Graham Neubig and Kevin Duh. On the Elements of an Accurate Tree-to-String Machine Translation System. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–149. Association for Computational Linguistics, 2014. doi: 10.3115/v1/P14-2024. URL `http://www.aclweb.org/anthology/P14-2024`.

[258] Sonja Nieflen and Hermann Ney. Improving SMT quality with morpho-syntactic analysis. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, 2000. URL `http://www.aclweb.org/anthology/C00-2162`.

[259] Vassilina Nikoulina and Marc Dymetman. Using Syntactic Coupling Features for Discriminating Phrase-Based Translations (WMT-08 Shared Translation Task). In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 159–162. Association for Computational Linguistics, 2008. URL `http://www.aclweb.org/anthology/W08-0323`.

[260] Joakim Nivre. Dependency Grammar and Dependency Parsing. Technical Report MSI 05133, Växjö University, School of Mathematics and Systems Engineering, 2005. URL `http://stp.lingfil.uu.se/~nivre/docs/05133.pdf`.

[261] Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebirolu Eryiit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çar Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökrmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà M, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phng Lê Hng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lng Nguyn Th, Huyn Nguyn Th Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalnia, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh,

Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. Universal Dependencies 2.0, 2017. URL `http://hdl.handle.net/11234/1-1983`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

[262] Franz Josef Och and Hermann Ney. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4), 2004. URL `http://www.aclweb.org/anthology/J04-4002`.

[263] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72. Association for Computational Linguistics, 2014. doi: 10.3115/v1/S14-2008. URL `http://aclanthology.coli.uni-saarland.de/pdf/S/S14/S14-2008.pdf`.

[264] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926. Association for Computational Linguistics, 2015. doi: 10.18653/v1/S15-2153. URL `http://aclanthology.coli.uni-saarland.de/pdf/S/S15/S15-2153.pdf`.

[265] Sebastian Padó. *User's guide to `sigf`: Significance testing by approximate randomisation*, 2006. `https://www.nlpado.de/~sebastian/software/sigf.shtml`.

[266] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[267] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv preprint arXiv:1605.07277*, 2016.

[268] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting Adversarial Input Sequences for Recurrent Neural Networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE, 2016.

[269] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks Against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4944-4. doi: 10.1145/3052973.3053009. URL http://doi.acm.org/10.1145/3052973.3053009.

[270] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. URL http://www.aclweb.org/anthology/P02-1040.

[271] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. *arXiv preprint arXiv:1802.08129*, 2018.

[272] Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland, 2014.

[273] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-Margin Tensor Neural Network for Chinese Word Segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1028.

[274] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations.

In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2018.

[275] Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. Evaluating Semantic Phenomena in Neural Machine Translation Using Natural Language Inference. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2018.

[276] Daniel Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, 2003.

[277] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[278] Peng Qian, Xipeng Qiu, and Xuanjing Huang. Analyzing Linguistic Knowledge in Sequential Model of Sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 826–835, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1079`.

[279] Peng Qian, Xipeng Qiu, and Xuanjing Huang. Investigating Language Universal and Specific Properties in Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1140`.

[280] Chris Quirk, Arul Menezes, and Colin Cherry. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279. Association for Computational Linguistics, 2005. URL `http://www.aclweb.org/anthology/P05-1034`.

[281] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages

2383–2392. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1264. URL http://www.aclweb.org/anthology/D16-1264.

[282] Gabrielle Ras, Pim Haselager, and Marcel van Gerven. Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges. *arXiv preprint arXiv:1803.07517*, 2018.

[283] Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1998.

[284] D Raj Reddy. Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave. *The Journal of the Acoustical Society of America*, 40(5):1273–1273, 1966.

[285] Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-4702.

[286] Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James Glass. Neural attention for learning to rank questions in community question answering. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1734–1745, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[287] Salvatore Romeo, Giovanni Da San Martino, Yonatan Belinkov, Alberto Barrn-Cedeo, Mohamed Eldesouki, Kareem Darwish, Hamdy Mubarak, James Glass, and Alessandro Moschitti. Language processing and learning models for community question answering in arabic. *Information Processing & Management*, August 2017. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2017.07.003. URL http://www.sciencedirect.com/science/article/pii/S0306457316306720.

[288] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[289] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Acoustics, Speech*

and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 4580–4584. IEEE, 2015.

[290] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the Speech Front-End with Raw Waveform CLDNNs. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[291] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, Ahmed Abdelali, Yonatan Belinkov, and Stephan Vogel. Challenging Language-Dependent Segmentation for Arabic: An Application to Machine Translation and Part-of-Speech Tagging. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[292] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *15th Annual Conference of the International Speech Communication Association (Interspeech)*, pages 338–342, 2014. URL `http://www.isca-speech.org/archive/interspeech_2014/i14_0338.html`.

[293] Haşim Sak, Félix de Chaumont Quitry, Tara Sainath, Kanishka Rao, et al. Acoustic Modelling with CD-CTC-SMBR LSTM RNNs. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 604–609. IEEE, 2015.

[294] Suranjana Samanta and Sameep Mehta. Towards Crafting Text Adversarial Samples. *arXiv preprint arXiv:1707.02812*, 2017.

[295] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

[296] Helmut Schmid. Part-of-Speech Tagging with Neural Networks. In *Proceedings of the 15th International Conference on Computational Linguistics (Coling 1994)*, pages 172–176, Kyoto, Japan, August 1994. Coling 1994 Organizing Committee.

[297] Helmut Schmid. LoPar: Design and Implementation. Bericht des Sonderforschungsbereiches “Sprachtheoretische Grundlagen fr die Computerlinguistik” 149, Institute for Computational Linguistics, University of Stuttgart, 2000.

[298] Holger Schwenk. Continuous Space Language Models. *Comput. Speech Lang.*, 21 (3):492–518, July 2007. ISSN 0885-2308. doi: 10.1016/j.csl.2006.09.003. URL `http://dx.doi.org/10.1016/j.csl.2006.09.003`.

[299] Holger Schwenk. Continuous Space Translation Models for Phrase-Based Statistical Machine Translation. In *Proceedings of COLING 2012: Posters*, pages 1071–1080. The COLING 2012 Organizing Committee, 2012. URL http://www.aclweb.org/anthology/C12-2104.

[300] Holger Schwenk, Daniel Dchelotte, and Jean-Luc Gauvain. Continuous Space Language Models for Statistical Machine Translation. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 723–730, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1273073.1273166.

[301] Rico Sennrich. How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/E17-2060.

[302] Rico Sennrich and Barry Haddow. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-2209. URL http://www.aclweb.org/anthology/W16-2209.

[303] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1005. URL http://www.aclweb.org/anthology/N16-1005.

[304] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1162.

[305] Libin Shen, Jinxi Xu, and Ralph Weischedel. String-to-Dependency Statistical Machine Translation. *Computational Linguistics*, 36(4):649–671, 2010.

[306] Xing Shi, Inkit Padhi, and Kevin Knight. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the 2016 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1159`.

[307] Stuart M. Shieber and Yves Schabes. Synchronous Tree-Adjoining Grammars. In *COLNG 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*, 1990. URL `http://www.aclweb.org/anthology/C90-3045`.

[308] Richard Socher. *Recursive Deep Learning for Natural Language Processing and Computer Vision*. PhD thesis, Stanford University, 2014.

[309] Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. In *Proceedings of NIPS Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

[310] Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. Parsing with Compositional Vector Grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P13-1045`.

[311] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1170`.

[312] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *TACL*, 2014.

[313] Hagen Soltau, Hank Liao, and Hasim Sak. Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition. *arXiv preprint arXiv:1610.09975*, 2016.

[314] Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. Syntactically Guided Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers), pages 299–305. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-2049. URL http://www.aclweb.org/anthology/P16-2049.

[315] David Stallard, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. Unsupervised Morphology Rivals Supervised Morphology for Arabic MT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, 2012.

[316] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B18WgG-CZ.

[317] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

[318] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[319] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[320] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. Multitask Learning with Low-Level Auxiliary Tasks for Encoder-Decoder Based Speech Recognition. In *Proc. Interspeech 2017*, pages 3532–3536, 2017. doi: 10.21437/Interspeech.2017-1118. URL http://dx.doi.org/10.21437/Interspeech.2017-1118.

[321] Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. Applying Morphology Generation Models to Machine Translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June 2008.

[322] Ke Tran, Arianna Bisazza, and Christof Monz. The Importance of Being Recurrent for Modeling Hierarchical Structure. *arXiv preprint arXiv:1803.03585*, 2018.

[323] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005. URL `http://www.aclweb.org/anthology/I05-3027`.

[324] Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1040`.

[325] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney. Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[326] Clara Vania and Adam Lopez. From Characters to Words to in Between: Do We Capture Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2016–2027. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1184. URL `http://www.aclweb.org/anthology/P17-1184`.

[327] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[328] S Veldhoen, D Hupkes, W Zuidema, et al. Diagnostic Classifiers: Revealing how Neural Networks Process Hierarchical Structure. In *CEUR Workshop Proceedings*, volume 1773, 2016.

[329] Sami Virpioja, Jaakko J. Vyrynen, Mathias Creutz, and Markus Sadeniemi. Morphology-Aware Statistical Machine Translation Based on Morphs Induced in an Unsupervised Manner. In *Proceedings of the Machine Translation Summit XI*, pages 491–498, 2007.

[330] Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. Word Representation Models for Morphologically Rich Languages in Neural Machine Translation. *arXiv preprint arXiv:1606.04217*, 2016.

[331] Alex Waibel, Ajay N. Jain, Arthur E. McNair, Hiroaki Saito, Alexander G. Hauptmann, and Joe Tebelskis. JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 793–796, April 1991. doi: 10.1109/ICASSP.1991.150456.

[332] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme Recognition Using Time-Delay Neural Networks. In *Readings in speech recognition*, pages 393–404. Elsevier, 1990.

[333] Shuai Wang, Yanmin Qian, and Kai Yu. What Does the Speaker Embedding Encode? In *Interspeech 2017*, pages 1497–1501, 2017. doi: 10.21437/Interspeech.2017-1125. URL `http://dx.doi.org/10.21437/Interspeech.2017-1125`.

[334] Ye-Yi Wang and Alex Waibel. Connectionist F-structure Transfer. *Recent Advances in Natural Language Processing: Selected Papers from RANLP'95*, 136:393, 1997.

[335] Yu-Hsuan Wang, Cheng-Tao Chung, and Hung-yi Lee. Gate Activation Signal Analysis for Gated Recurrent Neural Networks and Its Correlation with Phoneme Boundaries. In *Interspeech 2017*, 2017.

[336] Warren Weaver. Translation. *Machine translation of languages*, 14:15–23, 1955.

[337] Adrian Weller. Challenges for transparency. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2017.

[338] Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn. Syntax-based Statistical Machine Translation. *Synthesis Lectures on Human Language Technologies*, 9 (4):1–208, 2016.

[339] Dekai Wu and Pascale Fung. Semantic Roles for SMT: A Hybrid Two-Pass Model. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 13–16. Association for Computational Linguistics, 2009. URL `http://www.aclweb.org/anthology/N09-2004`.

[340] Dekai Wu and Pascale Fung. Can Semantic Role Labeling Improve SMT? In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 218–225. Universitat Politècnica de Catalunya, May 2009.

[341] Dekai Wu, Pascale N Fung, Marine Carpuat, Chi-kiu Lo, Yongsheng Yang, and Zhaojun Wu. Lexical Semantics for Statistical Machine Translation. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer Publishing Company, Incorporated, 2011.

[342] Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou. Sequence-to-Dependency Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 698–707. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1065. URL `http://www.aclweb.org/anthology/P17-1065`.

[343] Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. Extracting Pre-ordering Rules from Predicate-Argument Structures. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 29–37. Asian Federation of Natural Language Processing, 2011. URL `http://www.aclweb.org/anthology/I11-1004`.

[344] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv preprint arXiv:1609.08144*, 2016.

[345] Zhizheng Wu and Simon King. Investigating gated recurrent networks for speech synthesis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5140–5144. IEEE, 2016.

[346] Deyi Xiong, Min Zhang, and Haizhou Li. Modeling the Translation of Predicate-Argument Structure for SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911. Association for Computational Linguistics, 2012. URL `http://aclweb.org/anthology/P12-1095`.

[347] Kenji Yamada and Kevin Knight. A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, 2001. doi: 10.3115/1073012.1073079. URL `https://doi.org/10.3115/1073012.1073079`.

[348] Kenji Yamada and Kevin Knight. A Decoder for Syntax-based Statistical MT. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. URL `http://www.aclweb.org/anthology/P02-1039`.

[349] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. A Character-Aware Encoder for Neural Machine Translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3063–3070. The COLING 2016 Organizing Committee, 2016. URL `http://www.aclweb.org/anthology/C16-1288`.

[350] Victor H. Yngve. A framework for syntactic translation. In Sergei Nirenburg, Harold L. Somers, and Yorick A. Wilks, editors, *Readings in Machine Translation*, pages 39–44. MIT Press, 2003.

[351] Dong Yu and Deng Li. Summary and Future Directions. In *Automatic Speech Recognition: A Deep Learning Approach*, chapter 15. Springer-Verlag London, 1 edition, 2015.

[352] Dong Yu, Michael L Seltzer, Jinyu Li, Jui-Ting Huang, and Frank Seide. Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks. In *International Conference on Learning Representations (ICLR)*, 2013.

[353] Omar Zaidan, Jason Eisner, and Christine Piatko. Using “Annotator Rationales” to Improve Machine Learning for Text Categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267. Association for Computational Linguistics, 2007. URL `http://www.aclweb.org/anthology/N07-1033`.

[354] Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. Bilingually-constrained Phrase Embeddings for Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 111–121, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-1011`.

[355] Min Zhang, Hongfei Jiang, AiTi Aw, Jun Sun, Sheng Li, and Chew Lim Tan. A Tree-to-Tree Alignment-based Model for Statistical Machine Translation. *MT-Summit-07*, pages 535–542, 2007.

[356] Quan-shi Zhang and Song-chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, Jan 2018. ISSN 2095-9230. doi: 10.1631/FITEE.1700808. URL `https://doi.org/10.1631/FITEE.1700808`.

[357] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc., 2015.

[358] Ye Zhang, Iain Marshall, and Byron C. Wallace. Rationale-Augmented Convolutional Neural Networks for Text Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 795–804. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1076. URL `http://www.aclweb.org/anthology/D16-1076`.

[359] Yu Zhang. *Exploring Neural Network Architectures For Acoustic Modeling*. PhD thesis, Massachusetts Institute of Technology, September 2017.

[360] Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849. IEEE, 2017.

[361] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 4:371–383, 2016. ISSN 2307-387X. URL `https://transacl.org/ojs/index.php/tacl/article/view/863`.

[362] Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The United Nations Parallel Corpus v1.0. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

[363] Victor Zue, James Glass, Michael Phillips, and Stephanie Seneff. The MIT SUMMIT Speech Recognition System: A Progress Report. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '89, pages 179–189, Stroudsburg, PA, USA, 1989. Association for Computational Linguistics. doi: 10.3115/100964.100983. URL `https://doi.org/10.3115/100964.100983`.