Representation Learning of Linguistic Structures with Neural Networks

Kazuya Kawakami

Merton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2021

Acknowledgements

I am grateful to have an opportunity to pursue my research with the support of great advisors, colleagues, friends and family. Here, I would like to express my gratitude to the people who helped me complete this dissertation. I want to begin by acknowledging the invaluable support of my advisors, Phil Blunsom and Chris Dyer. Phil provided me with the perfect balance of guidance and freedom to explore new ideas. He always offered ideas and perspectives that I did not even think of. His expertise and kindness created great research environments that I enjoyed in Oxford and DeepMind. Chris has been my advisor since I started my graduate study at CMU. He always listened to my ideas and helped me develop them with his immense knowledge of machine learning and linguistics. Beyond technical advice, he always supported me with his passion and determination to deliver good research. I am very fortunate to work with both of my advisors. I am also thankful to Varun Kanade and Kyunghyun Cho for being my examiners and their helpful comments on improving this thesis.

I thank my colleagues at CMU, Oxford and DeepMind, who showed me that anything is possible when brilliant minds work hard together. Over the past ten years, machine learning research has progressed significantly, and my colleagues were always at the centre of the advance. I was so motivated by their achievements and proud to work with them. I also appreciate the financial support from the Funai scholarship and the Ripplewood scholarship. Their generous support allowed me to study in the great schools far away from home.

Most especially, I would thank my parents, Yoshiaki and Eiko, and my brother Tomoya. None of this would have been possible without their support and encouragement. Thank you very much for everything.

Contents

List of Figures

Chapter 1

Introduction

Language is an open-ended system that can represent an infinite set of ideas with a finite set of linguistic units (von Humboldt, 1836; Chomsky, 1965). To be able to express ideas about an evolving world, it continuously accommodates new concepts with the creations of words and expressions. Since the number of linguistic observations that an individual can experience is limited, a learner (a human or a machine) needs to be able to efficiently generalize to unseen words and sentences. Linguistic generalization is made possible by recombining a small number of atomic units, such as phonemes, morphemes and words, according to grammars that define how these linguistic structures may combine and how they should be interpreted. The mechanism of how a learner is able to discover, represent and use such structures latent in audio and text has been studied for a long time in linguistics, cognitive science and machine learning.

From the machine learning perspective, the problem of learning to discover and represent structure is in the category of unsupervised learning which aims at learning meaningful representations through the modelling of data distributions. In this thesis, I refer to **explicit modelling** as a method that represents structures as explicit latent variables. The latent variables can be aligned with linguistic representations such as morphemes, words and grammars. The combinatorial nature of discrete variables enables the model to efficiently generalize to unseen observations. For example, if the model knows about the word "apple" and a plural marker "-s", it is possible to represent "apples" as a combination of "apple" and "-s" instead of treating it as a new word. Moreover, the plural marker can be reused to represent other plural words such as "dog-s" and "cat-s". **Implicit modelling** expects to learn structures implicitly using high-dimensional vector representations, namely

distributed representations (Hinton, 1984), and functions to compose the vectors such as neural networks. The combination of high-dimensional vector representations and functions represents complex interactions between variables and generalizes to unseen observations. When a model has learned a vector representation of "apple", it is possible to construct the representation of "apples" by changing some elements of the vector that correspond to plurality instead of learning a new vector from scratch. Similarly to the example of explicit modelling, the transformation to represent plurality can be reused for other plural words. Although the explicit and implicit modelling seems to be different at a glance, they share the same goal, that is to implement efficient generalization mechanisms in computational models.

In practice, there is a trade-off between the explicit and implicit models in terms of interpretability and scalability. Explicit models, which define generative processes of the data, are able to use human knowledge about the data in the modelling assumptions and in prior distributions. The provided knowledge enable the models to learn efficiently from a small amount of data. And the probabilistic framework provides interpretations about how a model induced structures from the input. However, since explicit models need to consider exponentially large combinations of variables, they are computationally expensive and can be difficult to scale to large corpora. On the other hand, implicit models, that define differentiable computation graphs, can learn efficiently from large-scale data using gradient-based optimizations. However, since the implicit models expect to learn structures directly from the data, they often require a large amount of data to discover meaningful structures. Also, the structures learned in high-dimensional vectors are not directly useful for interpretation. Although implicit models, such as neural networks, are successful in many practical applications where large-scale data is available such as text classification, question answering and translation, there is a lot of room for improvements in terms of sample efficiency and interpretability.

1.1 Goals and Approaches

The goal of this thesis is to explore whether it is possible to mitigate the trade-off between the explicit and implicit modelling and design methods to incorporate linguistic structures while taking advantage of the representational power of distributed representations. The

combinations of explicit and implicit modelling place this thesis into a unique position to explore diverse questions at the intersection of probabilistic modelling, neural networks and language acquisition including modelling, learning, inference and evaluation. Moreover, the combination opens new research directions to investigate the relationship between linguistic representations learning and other continuous observations such as audio, vision and sensorimotor experiences. Since language ultimately must communicate about experience in different modalities, this link is essential.

In the first part of the thesis, I explore the explicit modelling of word creation and reuse in the context of open-vocabulary language modelling. I propose a neural network augmented with a hierarchical structure and a memory component that explicitly models the generation of new words and supports the frequent reuse of new words. The research question is whether the explicit modelling assumption is useful for improving the performance of language modelling compared to the implicit model without using any linguistic structures. The model is evaluated in terms of language modelling performance (i.e. held-out perplexity) on typologically diverse languages and compared with a character-level neural language model which does not explicitly represent any linguistic structure. The results show that the proposed explicit model improve the performance on language modelling in all tested languages and analysis demonstrates that the model is able to use the memory architecture appropriately.

In the second part, I extend the open-vocabulary language model to discover word-like structures without any supervision of word boundaries. In contrast to previous work on word segmentation and language modelling that focuses only on either structure discovery or language modelling, the hypothesis is that it is possible to learn good predictive distributions of language at the same time as discovering good explicit structures. Thus, the proposed model combines the benefit of explicit and implicit modelling by parameterizing an explicit probabilistic model using neural networks. The proposal includes a differential learning algorithm that efficiently marginalizes all possible segmentation decisions and a regularization method that is crucial for successful structure induction and language modelling. The model is evaluated in terms of both language modelling performance (i.e. held-out perplexity) and the quality of induced word structures (i.e. precision metrics compared to the human reference). The results show that the proposed model improves language modelling performance over neural language models and discovers word-like units better than Bayesian

word segmentation models. Moreover, conditioning on visual context improves performance on both.

In the last part, I present a method to discover acoustic structures implicitly from raw audio signals and show that the model can learn useful representations from large-scale, real-world data. The aim is to learn representations that are robust to domain shifts (e.g. read English to spoken English) and generalize well to many languages. Since the structures are not induced explicitly, the representations are evaluated based on the impact on downstream speech recognition tasks which predicts phonetic structure in utterances. The results show that the representations learned from diverse and noisy data provide significant improvements on speech recognition tasks in terms of performance, data efficiency and robustness. Moreover, the representations generalize well to many languages including tonal and low-resource languages.

1.2 Thesis Outline

Chapter 2: Background In this chapter, I review major concepts that underlie this thesis. I first describe the field of representation learning starting from the qualities that the representations should have. I then summarize different modelling options and techniques proposed in previous works. Secondly, I review linguistic structures studied in linguistics such as phonemes, morphemes and words and the related machine learning literature. Lastly, I review the field of language acquisition that studies how linguistic structures emerge in the human brain and how language learning is related to non-linguistic referents and our sensorimotor system.

Chapter 3: Word Creation and Reuse This chapter introduces an open-vocabulary language model that is able to **explicitly** incorporate the creation and the reuse of new words. Unlike closed vocabulary language models which ignore less frequent words in a corpus, open-vocabulary models need to capture the long tail in the power-law distribution that natural language follows (Zipf, 1949). Moreover, there is a phenomenon, called burstiness (Church and Gale, 1995; Church, 2000), that rare words appear many times in a single document (in burst). In order to capture these properties, I propose a model that exploits explicit word-level structure. The model represents a word as a sequence of characters tokenized by space tokens and generates words with two processes, one is to

generate words character-by-character and the other is to generate words from a memory that stores recently observed words. The experiments show that the model learned better predictive distributions over 7 typologically diverse languages. One limitation of the model is that it is not applicable to some languages, such as Japanese and Chinese, which do not use explicit space symbols to tokenize words. Thus, in the next chapter, I extend the model to discover words-like units at the same time as learning predictive distributions over character sequences.

Chapter 4: Word Discovery and Grounding In this section, I present a segmental neural language model that discovers word-like units **explicitly** from unsegmented character sequences at the same time as learning predictive distributions over the characters. In contrast to the previous segmentation models that treat word segmentation as an isolated task (e.g. Bayesian language models), the model aims to solve word discovery and language modelling without losing performance on both tasks. Moreover, I investigate the potential of incorporating non-linguistic visual context during language learning. Experiments show that the unconditional model learns predictive distributions better than character LSTM models, discovers words competitively with nonparametric Bayesian word segmentation models, and that modelling language conditional on visual context improves performance on both.

Chapter 5: Acoustic Modelling In this chapter, I propose a method to discover acoustic representations, which correlate with phonetic structures, **implicitly** from continuous raw audio signals. Unlike previous works that have been focused on evaluating the representations in terms of their ability to improve the performance of speech recognition systems on read English (e.g. Wall Street Journal and LibriSpeech), we learn representations from up to 8000 hours of diverse and noisy speech data and evaluate the representations by looking at their robustness to domain shifts and their ability to improve recognition performance in many languages. The results show that the representations confer significant robustness advantages to the resulting recognition systems: we see significant improvements in out-of-domain transfer relative to baseline feature sets and the features likewise provide improvements in 25 phonetically diverse languages.

Chapter 6: Conclusion To conclude this thesis, I return to the original question of how machine learning models can efficiently generalize far beyond their observations by leveraging linguistic structures as humans do. I summarize the benefits and drawbacks of

different modelling options by reviewing the results and findings in the previous chapters. I then discuss how the proposed methods can be combined to develop a single model that learns languages using various linguistic structures in different modalities (i.e. audio, text and vision etc.). I present several future directions and conclude.

1.3 Research Contributions

This thesis contains the following research contributions.

- I propose a character-level open-vocabulary neural language model that handles the creation and reuse of new word types using explicit word segmentation. I show that the caching mechanism in the language model enables us to learn a better predictive distribution of language and capture the "bursty" distribution of words.

- I construct a new open-vocabulary language modelling corpus (the Multilingual Wikipedia Corpus; MWC) from comparable Wikipedia articles in 7 typologically diverse languages and demonstrate the effectiveness of the proposed model across this range of languages. I made the dataset publicly available and there are many follow up works with the dataset.

- I propose a segmental neural language model that explicitly discovers and represents linguistic units latent in unsegmented character sequences. I find that the proposed lexical memory component provides a good bias for the model to discover words and to learn predictive distributions. Experiments show that the model outperforms character LSTM models, discovers words competitively with nonparametric word segmentation models.

- I proposed a differentiable prior that regularizes the segmental neural language model based on the expectation of the length of each segment. The prior can be computed efficiently using the above dynamic programming algorithm under the expectation semiring. The results show that the prior is necessary for inducing good word-like structures.

- I extend the segmental neural language model to be able to condition on side information, enabling not just the discovery of word-like units based on statistical regularities

in the training sequences but also based on associations with nonlinguistic visual context.

- I show that the representations grounded to visual context improve performance on language modelling and word discovery. The large-scale experiments on word discovery and visual grounding are new. I constructed a dataset for grounded word segmentation based on the MS-COCO dataset and made the dataset available for follow up works.

- I present an extension of the unsupervised method for learning speech representations that implicitly discovers phonetic structure from large-scale corpora of unlabelled raw audio signals.

- I show that existing acoustic features, MFCC and Log-filter bank, are not robust to domain shifts (e.g. from read English speech to spoken English speech).

- I show that the representations confer significant sample-efficiency and robustness advantages to the resulting speech recognition systems. I show that the acoustic representations provide improvements in 25 phonetically diverse languages. The trained model is shared with other research teams and used as an audio feature extractor.

Chapter 2

Background

2.1 Representation Learning

Representation is the mathematical expression of observations such as image, audio and text. Machine learning models take the representation as input and learn to exploit structures in the data to predict an output. In theory, expressive models, with an infinite amount of data, should be able to approximate any mapping function from input to output. However, since there is a limit to the amount of data and parameters to build a model, it is important to use representations that already encode relevant information about the data in order to find a good solution with a limited amount of data. Traditionally, human experts designed pre-processing and transformation methods for various data types. The process of designing representations involves empirical observations about the data and scientific knowledge about human perception. For example, in image representations, HOG (histogram of oriented gradients) and SIFT (Scaled Invariance Feature Transform) features, which capture local patterns (edges or patches) in an image, have been widely used in practical applications. For audio representations, MFCC (Mel-frequency cepstral coefficients), the short-term power spectrum of a sound, is designed to approximate the nonlinear response of the human auditory system to frequency features. For text representation, bag-of-words representation, which represents text as a set of words without incorporating word order or grammatical structures, have been used as a de-facto standard. Representations of words often have used morphological features, and features derived from lexicons.

As machine learning models are applied to real-world problems that require to exploit complex factors latent in the data, representation learning, which aim at learning

representations directly from the data, became increasingly important.

2.1.1 Quality of Representations

The desiderata of representation learning are to discover latent structures in the data and to encode the structures into a useful form for training another model or analyzing the learned structures. Thus, the quality of representations is often characterized in terms of efficiency, generalization and interpretability.

Efficiency , Efficiency refers to the amount of labelled data and model capacity required to train a model on a downstream task. The hypothesis is that good representations should capture meaningful structures in a form that another model can exploit with a small amount of data and parameters. The efficiency is often studied in the context of semi-supervised learning and few-shot (low-resource) learning where labels for the task of interest are scarce. For example, Hénaff et al. (2020) trained image representations on an unlabelled image dataset and evaluated their representations in terms of the performance of a linear classifier trained on top of their representations and the amount of labelled data required to achieve good performance. Recently, Brown et al. (2020) showed that representations learned by a language model trained on a large-scale corpus are useful for solving diverse language processing tasks such as question answering and translation with zero or few labelled data.

Generalization , Generalization is about the robustness and transferability of the learned representations. Robust representations, which are insensitive to irrelevant regularities in the data, generalize well to data distributions that are different from the training distribution. The irrelevant regularities can be lighting conditions in images and noise conditions in audio. Moreover, robustness to adversarial perturbations, which are designed to fool machine learning models, is extensively studied in the field of adversarial machine learning for privacy and security reasons (Goodfellow et al., 2015; Kurakin et al., 2016). Good representations make adversarial attacks less likely to succeed. Transferability of representations is the ability to share the learned structures across different tasks. If representation learning algorithms capture generic knowledge, which is not task-specific but would be useful for different tasks, it is possible to solve many tasks without training representations for each task. Transferability is important not only for practical applications but also for developing general intelligence which solves diverse tasks with a single model. There are

many empirical results that support the success of learning transferable representations. For example in computer vision, it is common to use representations trained on image recognition tasks for other tasks such as object detection and semantic segmentation (Simonyan and Zisserman, 2015; He et al., 2016). In language, vector representations of words (Mikolov et al., 2013; Devlin et al., 2019) trained to predict surrounding context are able to achieve good performance on a set of tasks such as sentiment analysis and paraphrase detection in diverse domains (Wang et al., 2019).

Interpretability , Real-word data arise from complex interactions of many factors. For example, speech sounds contain variations from language, speaker and background noise etc. Ideally, we would like a representation learning algorithm to disentangle such explanatory factors and store them in an interpretable form in order to provide us with a better understanding of the data and the learning algorithm. Implicit vector representations can be evaluated qualitatively using dimensionality reduction or clustering techniques such as PCA (Hotelling, 1933) and t-SNE (Van der Maaten and Hinton, 2008). Also, it is possible to compare explicit representations with reference structures annotated by humans (Tsvetkov et al., 2015). In language, learned representations are often evaluated in terms of correlation with human-annotated semantic and syntactic structures (e.g. word similarity, word segmentation and grammar induction) in order to compare the learning mechanism of humans and algorithms. Moreover, there are attempts to use the disentangled factors for controlled text generation and style transfer (Bowman et al., 2016; Karras et al., 2019).

2.1.2 Learning Objective

One of the challenges of representation learning is to find learning objectives to discover meaningful representations without using labels for the task of interest. The learning objective needs to lead a model to retain relevant information about the data at the same time as excluding irrelevant regularities.

Supervised Learning Supervised learning is a way to learn representations using a supervisory signal from labelled data. The learned representation can be directly used to solve the task of interest and they can be reused to solve closely related tasks. For example, in computer vision, it is common to learn representations from a supervised image recognition

task (e.g. on ImageNet) and reused the representations for other tasks such as object detection and semantic segmentation (Simonyan and Zisserman, 2015; He et al., 2016). However, supervised learning often requires a large amount of annotated data specifically collected for each task. Since the process of creating labelled data is often expensive and time consuming, there is far less data available for supervised learning compared to the amount of unlabelled data. Also, for the case of reusing the representations, it is not straightforward to find a supervised learning task well aligned with downstream tasks. Thus, research on unsupervised (self-supervised) learning has been focused on learning representations from unlabelled data by designing learning signals that incorporate our assumptions about the properties that representations should have.

Unsupervised Learning In order to retain information in the data without supervision, information-theoretic objectives such as likelihood and mutual information are widely used. The maximum likelihood objective, which learns to maximize the likelihood of the data, aim at maximizing the amount of information stored in the representations. It is widely used to represent images (Salakhutdinov and Hinton, 2009; Van Oord et al., 2016), audio (Jaitly and Hinton, 2011; Oord et al., 2016) and language (Shannon, 1948; Bengio et al., 2003). The mutual information maximization technique, which explicitly maximizes the mutual information between data and representations, became popular in recent works on self-supervised learning (Chen et al., 2016; Hjelm et al., 2019; van den Oord et al., 2018). There are other objectives which learn to represent data distributions such as score matching (Hyvärinen and Dayan, 2005), auto-encoding (Baldi and Hornik, 1989; Hinton, 1990) and adversarial objectives (Goodfellow et al., 2014). Also, metric learning objectives which learn a distance function between different data are proposed to capture useful variations in the data (Bromley et al., 1993; Chopra et al., 2005; Hadsell et al., 2006; Chen et al., 2020).

Regularization (Prior) Regularization techniques play an important role to lead learning algorithms to discover meaningful structures and to exclude irrelevant regularities from the representations. For probabilistic modelling, it is possible to specify the prior distribution over latent variables which defines the probability of unobserved structures independent from observations. The prior distribution prevents a model from finding a trivial solution that perfectly reconstructs input observations without generalization. For non-probabilistic modelling, it is possible to use regularization techniques such as L1 regularization for

sparsity and L2 regularization and dropout (Srivastava et al., 2014) for countering overfitting. Moreover, a denoising criterion, which learns to discard specific variations that are irrelevant to the task of interest (e.g. sensitivity to rotation for object detection), is commonly used to implement desired invariances to the representations. For instance, the denoising auto-encoder (Vincent et al., 2010) is trained to discard random noise at the same time as reconstructing the input. In recent works, diverse noise conditions and synthetic transformations are used to implement invariances to the representations. Chen et al. (2020) show that image representations, which are trained to discard synthetic transformations, such as cropping, resizing, colour distortions, and Gaussian blur, are useful for image classification tasks.

2.1.3 Inductive Bias

Inductive bias is a set of explicit and implicit modelling assumptions that determine the form and the properties of learned representations in combination with the data. The choice of modelling assumptions influences the difficulty of learning and inference. In this thesis, the distinction between explicit and implicit modelling refers to the modelling assumptions and their generalization mechanism rather than the form of input in order to highlight the differences in learning and inference. For instance, character-level and word-level neural language models with recurrent neural networks both use discrete linguistic units as inputs (characters and words) and they explicitly assign a vector representation for each unit. However, these models do not have explicit modelling assumptions to discover or exploit linguistic structures and they expect to generalize to unseen examples using implicit vector representations instead of using explicit combination of categorical units (e.g. dog-s, cat-s). Thus, in this thesis, I refer to such models as implicit models. In contrast, the hybrid models proposed in this thesis have explicit modelling assumptions for word creation and reuse (§3) or word discovery (§4) even though the models expect to learn higher-level structures such as syntax and semantics implicitly with neural networks. Table 2.1 summarizes explicit and implicit linguistic representations in language models.

Explicit modelling Explicit modelling, which specifies the structure of representations as latent variables, provide probabilistic interpretations about structures discovered by the learning algorithm. Discrete structures, such as words (Goldwater et al., 2009), grammars (Johnson et al., 2007) and topic (Blei et al., 2003), are often treated as explicit latent

Input	Model	Linguistic Structure		
		Word	Syntax	Semantics
Character	Bayesian LM (Goldwater et al., 2009)	**Explicit**	Implicit	Implicit
	RNN LM (Mikolov et al., 2010)	Implicit	Implicit	Implicit
	Transformer LM (Vaswani et al., 2017)	Implicit	Implicit	Implicit
	Segmental LM (This work, §4)	**Explicit**	Implicit	Implicit
Word	RNN LM (Mikolov et al., 2010)	-*	Implicit	Implicit
	RNNG (Dyer et al., 2016)	-*	**Explicit**	Implicit

Table 2.1: Summary of explicit and implicit linguistic representations in language models. *Word-level models explicitly assign vector representations for words but the word structure is given as inputs.

variables. The combination of discrete structures enables the model to represent exponentially large space with limited number of variables. For instance, if the model knows about the word "apple" and a plural marker "-s", it is possible to represent "apples" as a combination of "apple" and "-s" instead of treating it as a new word. Moreover, the plural marker can be reused to represent other plural words such as "dog-s" and "cat-s". Also, if a model knows a rule to create a noun phrase with an adjective and a noun, it is possible to create almost infinite combinations of noun phrases such as "big dogs" and "big cats". The benefit of using explicit modelling is that the learned structures are easy to interpret and evaluate by comparing with well-studied linguistic structures. On the other hand, since it is expensive to consider all possible combinations of variables, the learning and inference of explicit models tend to be difficult. In order to avoid expensive exact inference, many approximate inference algorithms, such as a variational approximation (Jordan, 1998), Gibbs sampling (Geman and Geman, 1984), and Markov chain Monte Carlo (Kass et al., 1998; Jordan et al., 1999) have been proposed in previous works. However, in most cases, it is still expensive to learn from large-scale data. Recently, methods have been proposed to learn continuous latent variables with the variational Bayesian approach using expressive neural networks with stochastic gradient descent (Kingma and Welling, 2014). The variational method is successful at representing images, but it is still unclear whether it is possible to discover meaningful structures from sequential data.

Implicit modelling Implicit modelling expects to learn a mapping from data to representations using the implicit biases in the model. Neural networks are the common tool to express the non-linear mapping functions and to encode complex structures into distributed vector representations(Hinton, 1984). The implicit models expect to generalize to unseen observations using high-dimensional vectors and functions to combine the vectors. Although the complex dynamics in high-dimensional space is hard to analyse, I present some examples to show how the implicit models might generalize to unseen observations. For example, the word "apple" can be represented as a continuous vector with 100 dimensions. If the last dimension of the vector corresponds to plurality, it is possible to construct the representations of "apples" by changing the last element of the vector representations of "apple" without learning a new vector. Also, if there is a function that combines vector representations of an adjective and a noun to construct a vector representation of a noun phrase, it is possible to represent arbitrary combinations of an adjective and a noun. Although the representations are less interpretable compared to the explicit latent variables, the distributed representations are able to retain rich information about the data useful to solve downstream tasks. Also, since learning and inference tend to be cheaper than latent variable models, implicit modelling is often preferred for large-scale learning problems. For implicit modelling, inductive biases are in the model architecture and hyper-parameters. Convolutional neural networks (LeCun et al., 1989), recurrent neural networks (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997) and transformers (Vaswani et al., 2017) are all able to represent spatial and temporal structure in the data such as audio, video and text. However, it is important to select appropriate model architectures and hyper-parameters for successful representation learning.

2.2 Linguistic Structures

Language has different levels of representations such as phonemes, morphemes, words and grammars. Humans are able to combine the linguistic units to express and understand an infinite set of ideas with a limited set of building blocks: morphemes into a new word, and words into a new sentence. This section summarizes linguistic structures and discusses their connections with representation learning.

2.2.1 Connection with Representation Learning

In the previous section, I summarize the desired properties of representation learning in terms of efficiency, generalization and interpretability. Linguistic structures are well-defined representations that satisfy these requirements. For example, phonetic representations enable us to represent a continuous speech signal with a sequence of the phonetic alphabet by excluding irrelevant variations in speech such as speaker identity and noise condition. Phonetic representations enable us to efficiently analyze and recognize speech without processing raw speech signals from scratch. Morphology and syntax provide a set of rules to understand and generate words and sentences including new ones. The combination of categorical units (i.e. morphemes and words) is able to efficiently represent diverse use of language in an interpretable manner (e.g. phd-ing, googl-ing). As the examples show, the goal of studying linguistic structure is closely aligned with that of representation learning. In the following sections, I summarize how the different levels of linguistic units are used in human language and how the usage varies across different languages.

2.2.2 Phonemes

A phoneme is the smallest unit of speech sound perceived to have the same function to distinguish one word from another by speakers of a particular language. Phonetic analysis transcribes a continuous speech signal with a sequence of discrete symbols drawn from a set of the phonetic alphabet. For example, the English phonetic alphabet consists of 44 phonemes (20 vowels and 24 consonants) and Japanese has 24 phonemes (5 vowels, 16 consonants and 3 special moras).

Unlike morphemes and words that are already represented in text, the task of discovering and categorizing phonetic structures in speech sound requires abstracting variations in speech production such as duration, stress, pitch and intensity. Although humans are great at discovering such temporal structures and learning robust phonetic representations to recognize their language, the mechanism of learning such structures is still unclear. Computational analysis of phonetic structures often uses a special transformation of speech signals, called Mel-frequency cepstral coefficients (MFCC) which is specifically designed to incorporate the nonlinear response of the human auditory system to frequency features: humans are better at identifying small changes in a speech at lower frequencies.

Another distinction between phonemes and higher-level linguistic structures is the universality of the representation. Since the sounds that humans are able to produce are constrained to our vocal apparatus rather than the language-specific alphabet, acoustic representations can be shared across different languages. In fact, the International Phonetic Alphabet (IPA) is an attempt to represent all known languages using 107 sound symbols (consonants and vowels), 52 accents and 4 prosodic marks (Nicolaidis, 2005). However, since the phonetic systems vary considerably across languages, IPA has undergone many revisions since its creation.

2.2.3 Morphemes

A morpheme is the smallest meaningful unit in a language that links form with meaning. Morphemes can be classified into free or bound morphemes. Free morphemes can stand alone as a word (e.g. cat, dog). In other words, free morphemes cannot be divided into smaller parts and retain meaning. On the other hand, bound morphemes appear only as parts of words to modify the semantic meaning or the grammatical function of a word. Bound morphemes can be further classified into derivational and inflectional morphemes. Derivational morphemes change the semantic meaning or the part of speech of the affected word (e.g. teach-er, care-ful, social-ize, re-write). Inflectional morphemes modify the grammatical function of a word such as tense, gender and case, without changing the semantic meaning or the part of speech (e.g. cat-s, talk-ed, go-ing).

There are diverse ways of using morphological structures across different languages. Language can be classified into four categories based on the use of morphemes. An analytic language is a type of language where sentences are composed of free morphemes (i.e. little or no morphological change in words). Syntactic relations between morphemes and sentences are expressed using specific word types, word orders and particles rather than using morphological inflections. These include East Asian languages such as Vietnamese, Burmese and Classical Chinese. Agglutinating languages have words that consist of a linear chain of distinct morphemes (without changing spelling or phonetics) and each morpheme represents meaning. Examples of agglutinating languages include Finnish, Turkish, Swahili, Hungarian, Japanese etc. Unlike agglutinating language, where there is a one to one mapping from a morpheme to grammatical information, fusional languages use morphemes that represent multiple grammatical, syntactic, or semantic information. For example, in

French, conjugation is used to encode grammatical tense, person, gender and number with a single morphological alternation of a verb. Inflectional languages include major Indo-European languages (Sanskrit, Bengali, Greek, French, German etc.) and Balto-Slavic languages. Polysynthetic languages, such as Inuit and Siberian languages, show a high degree of morpheme per word ratio similarly to agglutinating and inflectional languages. However, unlike other language types, words in a polysynthetic language can contain multiple stems and often form long "sentence-words" which include subject and object nouns in a single verb.

In machine learning, it is common to use morphological analysis, which analyzes the internal structure of words, to reduce variations from morphological inflections in pre-processing. One of the most common methods is stemming which reduce inflected forms of a word to a common base form. For example, in English, inflected verbs "is", "am" and "are", are reduced to "be". However, since most of the morphological analyzers rely on language-specific rules which do not generalize across languages, there is a need to develop methods to represent structures inside of a word and relate them to higher-level structures incorporating the different use of morphemes.

2.2.4 Words

A word (lexeme) is a widely accepted linguistic unit that represents syntactic and semantic roles by itself. A lemma is the semantic meaning of a word which can be shared with a set of words that are related through the morphological inflections (e.g. a lemma go is shared with going, went and gone). As dictionaries list one or more meanings for each lemma, the meaning of a word can differ according to its context.

The vocabulary (lexicon) is a set of words in a language that constantly evolve by accommodating new concepts with creations of new words (Heaps, 1978). There are several ways to create new words. In order to represent a completely new concept in a language, it may be necessary to create a word type that is not related to existing words using the phonemes in the language (e.g. aspirin, Google). Also, it is possible to adopt a word from other languages without translation (e.g. anime, music). However, these methods are not productive in a sense that the existing knowledge about the language cannot be used to predict the meaning of new words. On the other hand, when there are related concepts in the language, existing words and morphemes can be reused to derive a new word. In

English, morphological derivations are commonly used. For example, a suffix -ology is often used to denote a field of study such as biology (bio + -ology), sociology (society + -ology). Compounding is another productive way to derive a word by combining two or more words to create a long word (snowball, railroad). Some languages such as German, Dutch, Swedish and Russian extensively use compounding. Conversions, which simply take a word and change its part of speech without extra morphemes, can be used. For instance, the noun Google can be used as a verb in modern English.

Representation learning of words has been studied for a long time in machine learning. In order to capture the complex semantic and grammatical properties of a word, a word is often represented as a high-dimensional continuous vector called a distributed representation. Automatic learning of word representations has relied on the distributional hypothesis: the meaning of a word is evidenced by the words that occur in its context (Harris, 1954). More precisely, the hypothesis suggests that similar words should have similar distributions of words in their surrounding context. Thus, most of the representation learning methods derive word meaning from co-occurrence statistics (Deerwester et al., 1990; Landauer and Dumais, 1997) or predictive distributions (Brown et al., 1992; Mikolov et al., 2013) learned from unlabeled corpora. Recent methods are designed to incorporate grammatical properties (Faruqui and Dyer, 2015), morphological structures (Cotterell et al., 2016) and multiple meanings (Neelakantan et al., 2015) of a word into representations.

2.2.5 Syntax

Syntax is a set of rules that governs how to fit words together to form a sentence. Since the grammatical rules are not observable, linguists have proposed various sets of rules to explain syntactic phenomena in many languages. Generally, the grammars can be grouped into constituency and dependency grammars.

Constituency grammars derive a sentence by recursively applying grammatical rules. For example, Fig.2.1 visualizes a derivation of a sentence with widely used constituency grammars, Context Free Grammars (CFG, Chomsky 1956) and Combinatory Categorial Grammars (CCG, Steedman 1996). As the example shows, a sentence is represented as a hierarchical organization of constituents. On the other hand, dependency grammars represent a sentence as a graph where the nodes are words in the sentence and the edges are grammatical relations between the words. Fig.2.1 shows an analysis of a sentence with

dependency grammars. Unlike constituency grammars which require a set of grammatical rules for each language, dependency grammars can be extended to other languages as it only requires a set of dependency relations between words. The Universal Dependencies (Nivre et al., 2016) provides an inventory of cross-linguistically applicable dependency relations. Both of the grammars are widely used in practical applications such as grammar checking, machine translation and question answering.

Computational models have been proposed to incorporate syntactic structures to represent phrases and sentences. Socher et al. (2011) proposed a neural network to learn vector representations of a sentence by recursively combining vector representations of words in the sentence following the deterministic tree structure provided by a supervised syntactic parser. More recently, neural networks that can infer and use deterministic tree structures have received a great deal of attention (Yogatama et al., 2016; Choi et al., 2018; Shen et al., 2018b). In contrast to the models that induce deterministic tree structures, Probabilistic context-free grammars (PCFG) and Recurrent Neural Network grammars (RNNG, Dyer et al. 2016; Kim et al. 2019) treat syntactic structures as explicit latent variables. These joint models of word sequences and syntactic structures are able to incorporate grammar ambiguity: two or more possible derivations for a sentence.

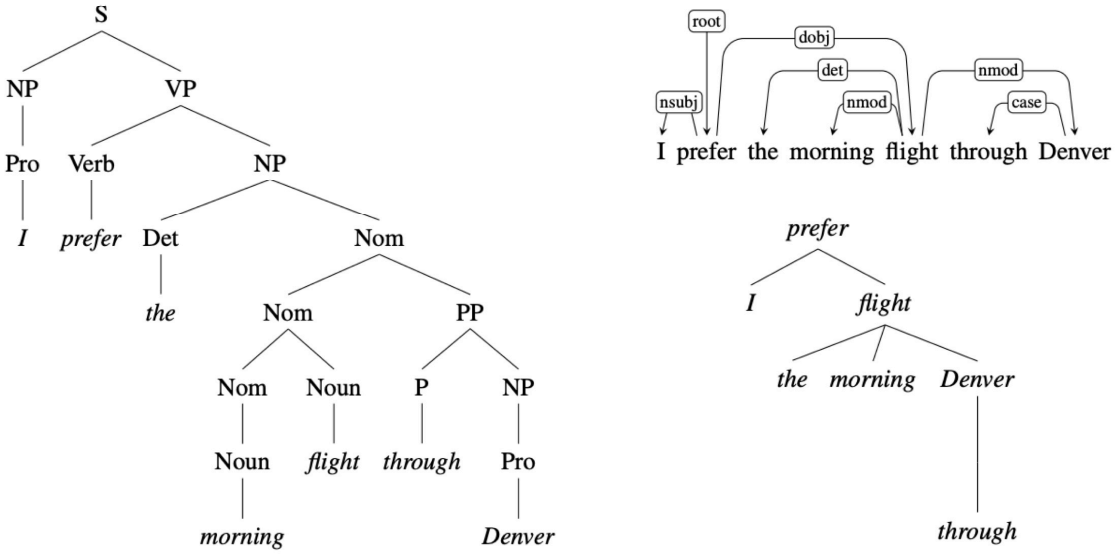

Figure 2.1: Visualization of parse tree of a sentence *I prefer the morning flight through Denver.* with Context-Free Grammar (left) and Dependency Grammars (right). Figures from Jurafsky and Martin (2020).

2.2.6 Semantics

Semantics is the most abstract linguistic structure that represents how the words are combined to form the meaning of a sentence. Semantic analysis, that translates natural language to a meaning representation, have been studied for programming languages, machine translation and question answering (Yngve et al., 1962; Woods, 1978; Alshawi, 1992; Copestake and Flickinger, 2000).

First-Order Logic represents the meaning of a sentence as a logical formula that enables us to use verification and inference to answer questions based on the meaning representations. For instance, the sentence "Bob doesn't like apples." can be represented as $\neg Like(Bob, apples)$ and it is possible to verify whether a question $Like(Bob, apples)$ is true or not. Minimal Recursion Semantics (MRS, Copestake et al. 1995) and Abstract Meaning Representation (AMR, Banarescu et al. 2013) are other meaning representation frameworks that represent the meaning of a sentence with a list and a graph structure respectively.

Although the explicit meaning representations are useful for verification, inference and interpretation, it is common to expect that such meaning representations are implicitly captured in distributed representations of a sentence. Kiros et al. (2015) proposed a method to learn vector representations of sentences by predicting surrounding context. The learned representations are used for sentence classification tasks. More recently, vector representations learned in neural language models are often used as semantic representations for downstream tasks such as sentence classification, translation and question answering (Devlin et al., 2019; Brown et al., 2020).

2.3 Language Acquisition

Language acquisition is a learning process to be able to perceive, comprehend and use language. Human infants start from discriminating and categorizing speech sounds from the language spoken around them. After that, they learn to recognize larger structures such as morphemes, words and sentences at the same time as associating meanings to these structures. Interestingly, humans seem to develop some sort of linguistic, both syntactic and semantic, representations that can be reused to understand and generate unseen words or sentences. However, it is unclear how such representations emerge in our brains.

2.3.1 Emergence of Linguistic Structure

Behaviourism (empiricism) is one of the hypotheses based on the framework provided by behaviourist psychology which postulates that all behaviour is learned through interaction with environments. Following the general framework, Skinner (1957) first proposed behaviourist's theory of language acquisition and argued that human linguistic behaviour is governed by the current features of the environment that the learner experiences and the history of reinforcement (i.e. rewards and punishments in response to linguistic behaviours) that the learner received.

Nativism (rationalism) is the other hypothesis that humans have an innate knowledge of various linguistic rules, constraints and principles instead of learning them from experience. For example, Chomsky (1980) suggested that humans must be born with an innate knowledge, known as Universal Grammar, based on his well-known argument, the "argument from the poverty of the stimulus", which states that the amount of linguistic inputs received by children is not enough to acquire detailed knowledge of their first language. The argument is widely accepted by modern linguists and applied to explain many linguistic phenomena in syntax, semantics and phonology.

Researchers in cognitive science and artificial intelligence started to use statistical methods to investigate the problem. Similarly to the discussion among linguists, there are two different approaches. Connectionists, who attempt to explain intellectual abilities with neural networks, suggests that highly structured linguistic knowledge, such as phonemes, morphemes, words, lexicon and grammars, can be learned and represented implicitly in distributed vector representations (Touretzky and Wheeler, 1989; Elman, 1991). On the other hand, computational linguists extensively investigated structured probabilistic models which implement explicit linguistic structures instead of learning them (Charniak, 1996; Manning and Schutze, 1999). The most recent probabilistic models employ the Bayesian framework to incorporate prior knowledge in statistical learning (Johnson et al., 2007; Goldwater et al., 2009).

2.3.2 Language Grounding

Language grounding is the broad question about how words and sentences get their meanings in relation to non-linguistic referents and our sensorimotor system. The problem, called the symbol grounding problem, is the central issue of many philosophical discussions

about machine intelligence. The famous Chinese room argument (Searle, 1980) states that machine intelligence that appears to understand language by perfectly processing symbols (e.g. Turing test, Turing and Haugeland 1950) cannot produce real understanding without grounding.

There are practical examples that show the need of language grounding for understanding. Frege (1892) pointed out that meaning is different from what it refers to. For example, the phrases, "the capital of England" and "the largest city in the UK" both refer to London even though the two phrases have completely different meanings. Humans seem to be able to pick the referent based on the knowledge acquired before but the process of learning such grounding is still unclear. Another example is the relationship between meaning and our sensorimotor systems such as taste, smell, touch, hearing, sight, and the sense of motion. The concepts related to the real-world environment such as position (e.g. the meaning of "above"), and size (e.g. an elephant is bigger than a squirrel) are hard to acquire just from text.

Recent machine learning research started to investigate the relationship between different modalities such as vision, motor control and 3D environment. Image captioning, which learns to generate a description of a given image, is one of the major fields studying the interaction between vision and language. Karpathy and Li (2015) proposed a method to generate a description by learning the alignment between objects in the image and phrases in its description. In an interactive learning setup an agent learns to understand and use language through interactions with the environment, explicitly implementing the behaviourists' hypothesis in the machine learning framework. Many environments have been considered such as game (Narasimhan et al., 2015; Yang et al., 2018), navigation (Hermann et al., 2017; Anderson et al., 2018) and robots (Thomason et al., 2015). However, since the learned representations are only evaluated on the specific environment, it is not clear whether the grounded representations are useful for other language processing tasks.

2.3.3 Computational models of Linguistic Structure Discovery

Statistical modeling of the process of linguistic structure discovery has been studied for a long time in structural linguistics, computational linguistics and cognitive science. The history traces back to the work of Harris (1955, 1968, 1970) which started from discovering morphological structures in a sequence of the phonetic alphabet based on the statistics in

the data. Further, the scientific evidence that human infants are able to discover word units by solely relying on the statistical regularities in continuous speech signals (Saffran et al., 1996) motivated the study of computational structure discovery as models of child language acquisition. There have been a lot of proposals over more than half a century of research for each linguistic structure such as phonology (Ellison, 1994), morphology (Goldsmith, 2001; Creutz and Lagus, 2002; Chahuneau et al., 2013a), word segmentation (De Marcken, 1995; Brent and Cartwright, 1996; Goldwater et al., 2009; Mochihashi et al., 2009), and syntax (Dowman, 2000; Kim et al., 2019).

2.4 Conclusion

In this section, I reviewed the concept of representation learning and linguistic structures. Representation learning is the field of study that aims to discover structures in the data and encode them into a useful form to train another machine learning model. I summarized the desiderata of good representations in terms of efficiency, generalization and interpretability. Linguistic structures have been studied to explain linguistic phenomena in human language. The combination of linguistic units enable us to efficiently represent diverse linguistic observations with a finite set of rules. I discussed the connections between the two fields in a sense that they both try to find representations that efficiently generalize to the new observations as humans do to acquire a language. In the following sections, I explore methods to combine the efficient generalization mechanism of representation learning and linguistic structures while incorporating the modeling challenges explained in this chapter.

Chapter 3

Word Creation and Reuse

3.1 Introduction

This chapter explores the explicit modelling of the word creation and reuse in the context of language modelling. Language modelling is an important problem in natural language processing with many practical applications (translation, speech recognition, spelling auto-correction, etc.). Recent advances in neural networks provide strong representational power to language models with distributed representations and unbounded dependencies based on recurrent networks (RNNs). However, most language models operate by generating words by sampling from a closed vocabulary which is composed of the most frequent words in a corpus. Rare tokens are typically replaced by a special token, called the unknown word token, $\langle \text{UNK} \rangle$. Although fixed-vocabulary language models have some important practical applications and are appealing models for study, they fail to capture two empirical facts about the distribution of words in natural languages. First, vocabularies keep growing as the number of documents in a corpus grows: new words are constantly being created (Heaps, 1978). Second, rare and newly created words often occur in "bursts", i.e., once a new or rare word has been used once in a document, it is often repeated (Church and Gale, 1995; Church, 2000).

The open-vocabulary problem can be solved by dispensing with word-level models in favor of models that predict sentences as sequences of characters (Sutskever et al., 2011;

The material in this chapter was presented in Kawakami et al. (2017).

Chung et al., 2017). Character-based models are quite successful at learning what (new) word forms look like (e.g., they learn a language's orthographic conventions that tell us that *sustinated* is a plausible English word and *bzoxqir* is not) and, when based on models that learn long-range dependencies such as RNNs, they can also be good models of how words fit together to form sentences.

However, existing character-sequence models have no explicit mechanism for modelling the fact that once a rare word is used, it is likely to be used again. In this section, I propose an extension to character-level language models that enables them to reuse previously generated tokens (§3.2). The starting point is a hierarchical LSTM that has been previously used for modelling sentences (word by word) in a conversation (Sordoni et al., 2015), except here I model words (character by character) in a sentence. To this model, I add a caching mechanism similar to recent proposals for caching that have been advocated for closed-vocabulary models (Merity et al., 2017; Grave et al., 2017). As word tokens are generated, they are placed in an LRU cache, and, at each time step the model decides whether to copy a previously generated word from the cache or to generate it from scratch, character by character. The decision of whether to use the cache or not is a latent variable that is marginalised during learning and inference. In summary, the model has three properties: it creates new words, it accounts for their burstiness using a cache, and, being based on LSTMs over word representations, it can model long range dependencies.

To evaluate the model, I perform ablation experiments with variants of the model without the cache or hierarchical structure. In addition to standard English data sets (PTB and WikiText-2), I introduce a new multilingual data set: the Multilingual Wikipedia Corpus (MWC), which is constructed from comparable articles from Wikipedia in 7 typologically diverse languages (§3.3) and show the effectiveness of the model in all languages (§3.4). By looking at the posterior probabilities of the generation mechanism (language model vs. cache) on held-out data, I find that the cache is used to generate "bursty" word types such as proper names, while numbers and generic content words are generated preferentially from the language model (§3.6).

3.2 Model

In this section, I describe the hierarchical character language model with a word cache. As is typical for RNN language models, the model uses the chain rule to decompose the problem into incremental predictions of the next word conditioned on the history:

$$p(\boldsymbol{w}) = \prod_{t=1}^{|\boldsymbol{w}|} p(w_t \mid \boldsymbol{w}_{<t}).$$

I make two modifications to the traditional RNN language model, which I describe in turn. First, I begin with a cache-less model I call the hierarchical character language model (HCLM; §3.2.1) which generates words as a sequence of characters and constructs a "word embedding" by encoding a character sequence with an LSTM (Ling et al., 2015). However, like conventional closed-vocabulary, word-based models, it is based on an LSTM that conditions on words represented by fixed-length vectors.[1]

The HCLM has no mechanism to reuse words that it has previously generated, so new forms will only be repeated with very low probability. However, since the HCLM is not merely generating sentences as a sequence of characters, but also segmenting them into words, I may add a word-based cache to which I add words keyed by the hidden state being used to generate them (§3.2.2). This cache mechanism is similar to the model proposed by Merity et al. (2017).

Notation. The model assigns probabilities to sequences of words $\boldsymbol{w} = w_1, \ldots, w_{|\boldsymbol{w}|}$, where $|\boldsymbol{w}|$ is the length, and where each word w_i is represented by a sequence of characters $\boldsymbol{c}_i = c_{i,1}, \ldots, c_{i,|\boldsymbol{c}_i|}$ of length $|\boldsymbol{c}_i|$.

3.2.1 Hierarchical Character-level Language Model

This hierarchical model satisfies the linguistic intuition that written language has (at least) two different units, characters and words.

The HCLM consists of four components, three LSTMs (Hochreiter and Schmidhuber, 1997): a character encoder, a word-level context encoder, and a character decoder (denoted

[1]The HCLM is an adaptation of the hierarchical recurrent encoder-decoder of (Sordoni et al., 2015) which was used to model dialog as a sequence of actions sentences which are themselves sequences of words. The original model was proposed to compose words into query sequences but I use it to compose characters into word sequences.

LSTM_{enc}, LSTM_{ctx}, and LSTM_{dec}, respectively), and a softmax output layer over the character vocabulary. Fig. 3.1 illustrates an unrolled HCLM.

$$p(\text{Pokémon}) = \lambda_t p_{lm}(\text{Pokémon}) + (1 - \lambda_t) p_{ptr}(\text{Pokémon})$$

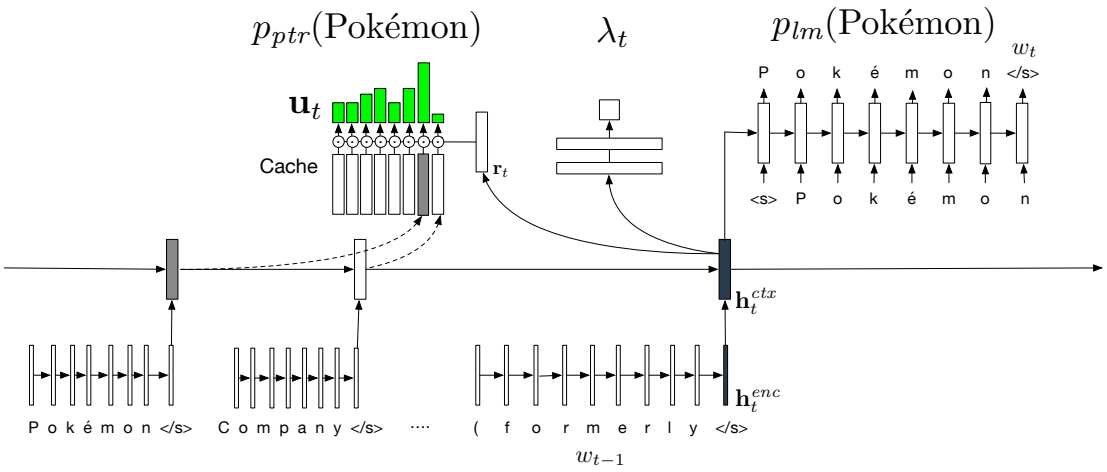

Figure 3.1: Description of Hierarchical Character Language Model with Cache.

Suppose the model reads word w_{t-1} and predicts the next word w_t. First, the model reads the character sequence representing the word $w_{t-1} = c_{t-1,1}, \ldots, c_{t-1,|\boldsymbol{c}_{t-1}|}$ where $|\boldsymbol{c}_{t-1}|$ is the length of the word generated at time $t-1$ in characters. Each character is represented as a vector $\mathbf{v}_{c_{t-1,1}}, \ldots, \mathbf{v}_{c_{t-1,|\boldsymbol{c}_{t-1}|}}$ and fed into the encoder LSTM_{enc} . The final hidden state of the encoder LSTM_{enc} is used as the vector representation of the previously generated word w_{t-1},

$$\mathbf{h}_t^{enc} = \mathrm{LSTM}_{enc}(\mathbf{v}_{c_{t-1,1}}, \ldots, \mathbf{v}_{c_{t-1,|\boldsymbol{c}_t|}}).$$

Then all the vector representations of words $(\mathbf{v}_{w_1}, \ldots, \mathbf{v}_{w_{|\boldsymbol{w}|}})$ are processed with a context LSTM_{ctx} . Each of the hidden states of the context LSTM_{ctx} are considered representations of the history of the word sequence.

$$\mathbf{h}_t^{ctx} = \mathrm{LSTM}_{ctx}(\mathbf{h}_1^{enc}, \ldots, \mathbf{h}_t^{enc})$$

Finally, the initial state of the decoder LSTM is set to be $\mathbf{h}_t^{ctx}$ and the decoder LSTM reads a vector representation of the start symbol $\mathbf{v}_{\langle \mathrm{s} \rangle}$ and generates the next word w_{t+1} character by character. To predict the j-th character in w_t, the decoder LSTM reads vector

representations of the previous characters in the word, conditioned on the context vector $\mathbf{h}_t^{ctx}$ and a start symbol.

$$\mathbf{h}_{t,j}^{dec} = \mathrm{LSTM}_{dec}(\mathbf{v}_{c_{t,1}}, \ldots, \mathbf{v}_{c_{t,j-1}}, \mathbf{h}_t^{ctx}, \mathbf{v}_{\langle \mathrm{S} \rangle}).$$

The character generation probability is defined by a softmax layer for the corresponding hidden representation of the decoder LSTM .

$$p(c_{t,j} \mid \boldsymbol{w}_{<t}, \boldsymbol{c}_{t,<j}) = \mathrm{softmax}(\mathbf{W}_{dec}\mathbf{h}_{t,j}^{dec} + \mathbf{b}_{dec})$$

Thus, a word generation probability from HCLM is defined as follows.

$$p_{lm}(w_t \mid \boldsymbol{w}_{<t}) = \prod_{j=1}^{|\boldsymbol{c}_t|} p(c_{t,j} \mid \boldsymbol{w}_{<t}, \boldsymbol{c}_{t,<j})$$

3.2.2 Continuous cache component

The cache component is an external memory structure which stores K elements of recent history. Similarly to the memory structure used in Grave et al. (2017), a word is added to a key-value memory after each generation of w_t. The key at position $i \in [1, K]$ is $\mathbf{k}_i$ and its value m_i. The memory slot is chosen as follows: if the w_t exists already in the memory, its key is updated (discussed below). Otherwise, if the memory is not full, an empty slot is chosen or the least recently used slot is overwritten. When writing a new word to memory, the key is the RNN representation that was used to generate the word ($\mathbf{h}_t$) and the value is the word itself (w_t). In the case when the word already exists in the cache at some position i, the $\mathbf{k}_i$ is updated to be the arithmetic average of $\mathbf{h}_t$ and the existing $\mathbf{k}_i$.

To define the copy probability from the cache at time t, a distribution over copy sites is defined using the attention mechanism of Bahdanau et al. (2015). To do so, I construct a query vector ($\mathbf{r}_t$) from the RNN's current hidden state $\mathbf{h}_t$,

$$\mathbf{r}_t = \tanh(\mathbf{W}_q\mathbf{h}_t + \mathbf{b}_q),$$

then, for each element i of the cache, a 'copy score,' $u_{i,t}$ is computed,

$$u_{i,t} = \mathbf{v}^T \tanh(\mathbf{W}_u\mathbf{k}_i + \mathbf{r}_t).$$

Finally, the probability of generating a word via the copying mechanism is:

$$p_{mem}(i \mid \mathbf{h}_t) = \mathrm{softmax}_i(\mathbf{u}_t)$$
$$p_{ptr}(w_t \mid \mathbf{h}_t) = p_{mem}(i \mid \mathbf{h}_t)[m_i = w_t],$$

where $[m_i = w_t]$ is 1 if the ith value in memory is w_t and 0 otherwise. Since p_{mem} defines a distribution of slots in the cache, p_{ptr} translates it into word space.

3.2.3 Character-level Neural Cache Language Model

The word probability $p(w_t \mid \boldsymbol{w}_{<t})$ is defined as a mixture of the following two probabilities. The first one is a language model probability, $p_{lm}(w_t \mid \boldsymbol{w}_{<t})$ and the other is pointer probability , $p_{ptr}(w_t \mid \boldsymbol{w}_{<t})$. The final probability $p(w_t \mid \boldsymbol{w}_{<t})$ is

$$\lambda_t p_{lm}(w_t \mid \boldsymbol{w}_{<t}) + (1 - \lambda_t) p_{ptr}(w_t \mid \boldsymbol{w}_{<t}),$$

where λ_t is computed by a multi-layer perceptron with two non-linear transformations using $\mathbf{h}_t$ as its input, followed by a transformation by the logistic sigmoid function:

$$\gamma_t = \mathrm{MLP}(\mathbf{h}_t), \qquad \lambda_t = \frac{1}{1 - e^{-\gamma_t}}.$$

I remark that Grave et al. (2017) use a clever trick to estimate the probability, λ_t of drawing from the LM by augmenting their (closed) vocabulary with a special symbol indicating that a copy should be used. This enables word types that are highly predictive in context to compete with the probability of a copy event. However, since I am working with an open vocabulary, this strategy is unavailable in the model, so I use the MLP formulation.

3.2.4 Training objective

The model parameters as well as the character projection parameters are jointly trained by maximizing the following log likelihood of the observed characters in the training corpus,

$$\mathcal{L} = -\sum \log p(w_t \mid \boldsymbol{w}_{<t}).$$

3.3 Datasets

I evaluate the model on a range of datasets, employing preexisting benchmarks for comparison to previous published results, and a new multilingual corpus which specifically tests the model's performance across a range of typological settings.

3.3.1 Penn Tree Bank (PTB)

I evaluate the model on the Penn Tree Bank. For fair comparison with previous works, I followed the standard preprocessing method used by Mikolov et al. (2010). In the standard preprocessing, tokenization is applied, words are lower-cased, and punctuation is removed. Also, less frequent words are replaced by unknown token (UNK),[2] constraining the word vocabulary size to be 10k. Because of this preprocessing, I do not expect this dataset to benefit from the modelling innovations I have introduced in the section. Fig.3.1 summarizes the corpus statistics.

	Train	Dev	Test
Character types	50	50	48
Word types	10000	6022	6049
OOV rate	-	**0.00%**	**0.00%**
Word tokens	0.9M	0.1M	0.1M
Characters	5.1M	0.4M	0.4M

Table 3.1: PTB Corpus Statistics.

3.3.2 WikiText-2

Merity et al. (2017) proposed the WikiText-2 Corpus as a new benchmark dataset. They pointed out that the preprocessed PTB is unrealistic for real language use in terms of word distribution. Since the vocabulary size is fixed to 10k, the word frequency does not exhibit a long tail. The WikiText-2 corpus is constructed from 720 articles. They provided two versions. The version for word level language modelling was preprocessed by discarding infrequent words. But, for character-level models, they provided raw documents without any removal of word or character types or lowercasing, but with tokenization. I make one change to this corpus: since Wikipedia articles make extensive use of characters from other languages; I replaced character types that occur fewer than 25 times with a dummy character

[2]When the unknown token is used in character-level model, it is treated as if it were a normal word (i.e. UNK is the sequence U, N, and K). This is somewhat surprising modelling choice, but it has become conventional (Chung et al., 2017).

(this plays the role of the ⟨UNK⟩ token in the character vocabulary). Tab. 3.2 summarizes the corpus statistics.

	Train	Dev	Test
Character types	255	128	138
Word types	76137	19813	21109
OOV rate	-	4.79%	5.87%
Word tokens	2.1M	0.2M	0.2M
Characters	10.9M	1.1M	1.3M

Table 3.2: WikiText-2 Corpus Statistics.

3.3.3 Multilingual Wikipedia Corpus (MWC)

Languages differ in what word formation processes they have. For character-level modelling it is therefore interesting to compare a model's performance across languages. Since there is at present no standard multilingual language modelling dataset, I created a new dataset, the Multilingual Wikipedia Corpus (MWC), a corpus of the same Wikipedia articles in 7 languages which manifest a range of morphological typologies. The MWC contains English (EN), French (FR), Spanish (ES), German (DE), Russian (RU), Czech (CS), and Finnish (FI).

To attempt to control for topic divergences across languages, every language's data consists of the same articles. Although these are only comparable (rather than true translations), this ensures that the corpus has a stable topic profile across languages.[3]

Construction & Preprocessing I constructed the MWC similarly to the WikiText-2 corpus. Articles were selected from Wikipedia in the 7 target languages. To keep the topic distribution to be approximately the same across the corpora, I extracted articles about entities which are explained in all the languages. I extracted articles which exist in all languages and each consists of more than 1,000 words, for a total of 797 articles. These

[3]The Multilingual Wikipedia Corpus (MWC) is available for download from `http://k-kawakami.com/research/mwc`

cross-lingual articles are, of course, not usually translations, but they tend to be comparable. This filtering ensures that the topic profile in each language is similar. Each language corpus is approximately the same size as the WikiText-2 corpus.

Wikipedia markup was removed with WikiExtractor,[4] to obtain plain text. I used the same thresholds to remove rare characters in the WikiText-2 corpus. No tokenization or other normalization (e.g., lowercasing) was done.

Statistics After the preprocessing described above, I randomly sampled 360 articles. The articles are split into 300, 30, 30 sets and the first 300 articles are used for training and the rest are used for dev and test respectively. Table 3.3 summarizes the corpus statistics.

	Char. Types			Word Types			OOV rate		Tokens			Characters		
	Train	Valid	Test	Train	Valid	Test	Valid	Test	Train	Valid	Test	Train	Valid	Test
EN	307	160	157	193808	38826	35093	6.60%	5.46%	2.5M	0.2M	0.2M	15.6M	1.5M	1.3M
FR	272	141	155	166354	34991	38323	6.70%	6.96%	2.0M	0.2M	0.2M	12.4M	1.3M	1.6M
DE	298	162	183	238703	40848	41962	7.07%	7.01%	1.9M	0.2M	0.2M	13.6M	1.2M	1.3M
ES	307	164	176	160574	31358	34999	6.61%	7.35%	1.8M	0.2M	0.2M	11.0M	1.0M	1.3M
CS	238	128	144	167886	23959	29638	5.06%	6.44%	0.9M	0.1M	0.1M	6.1M	0.4M	0.5M
FI	246	123	135	190595	32899	31109	8.33%	7.39%	0.7M	0.1M	0.1M	6.4M	0.7M	0.6M
RU	273	184	196	236834	46663	44772	7.76%	7.20%	1.3M	0.1M	0.1M	9.3M	1.0M	0.9M

Table 3.3: Summary of MWC Corpus.

Additionally, I show in Fig. 3.2 the distribution of frequencies of OOV word types (relative to the training set) in the dev+test portions of the corpus, which shows a power-law distribution, which is expected for the burstiness of rare words found in prior work. Curves look similar for all languages. Fig. 3.3 show distribution of frequencies of OOV word types in 6 languages.

3.4 Experiments

I now turn to a series of experiments to show the value of the hierarchical character-level cache language model. For each dataset I trained the model with LSTM units. To compare the results with a strong baseline, I also train a model without the cache.

[4] `https://github.com/attardi/wikiextractor`

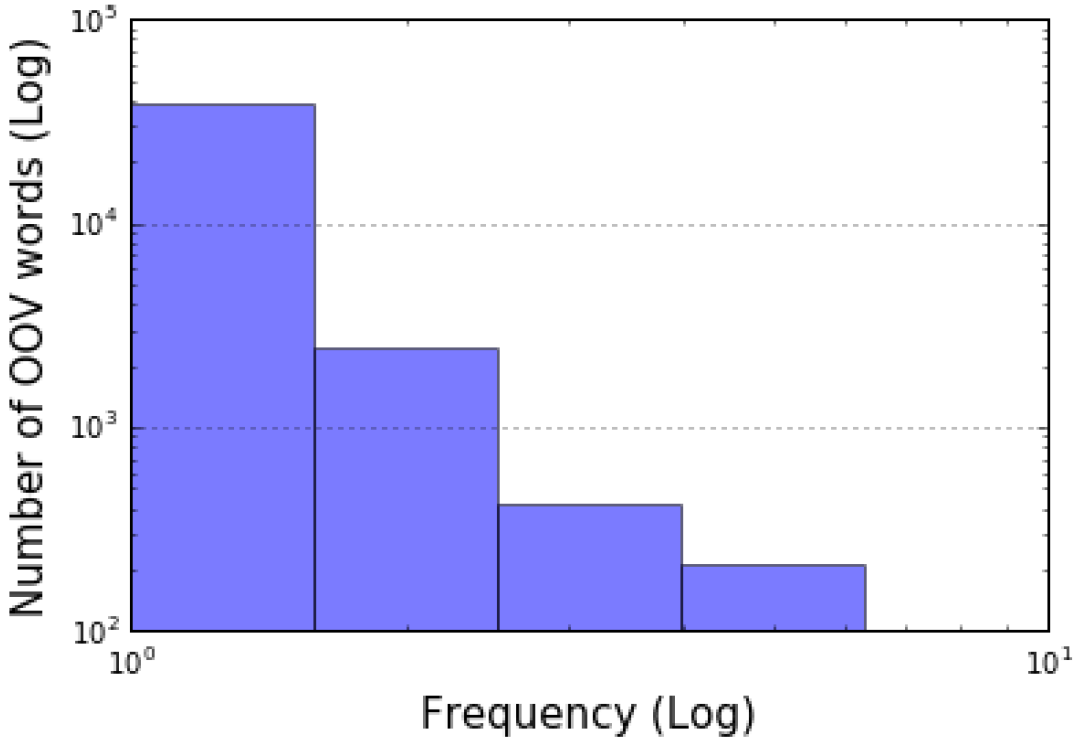

Figure 3.2: Histogram of OOV word frequencies in the dev+test part of the MWC Corpus (EN).

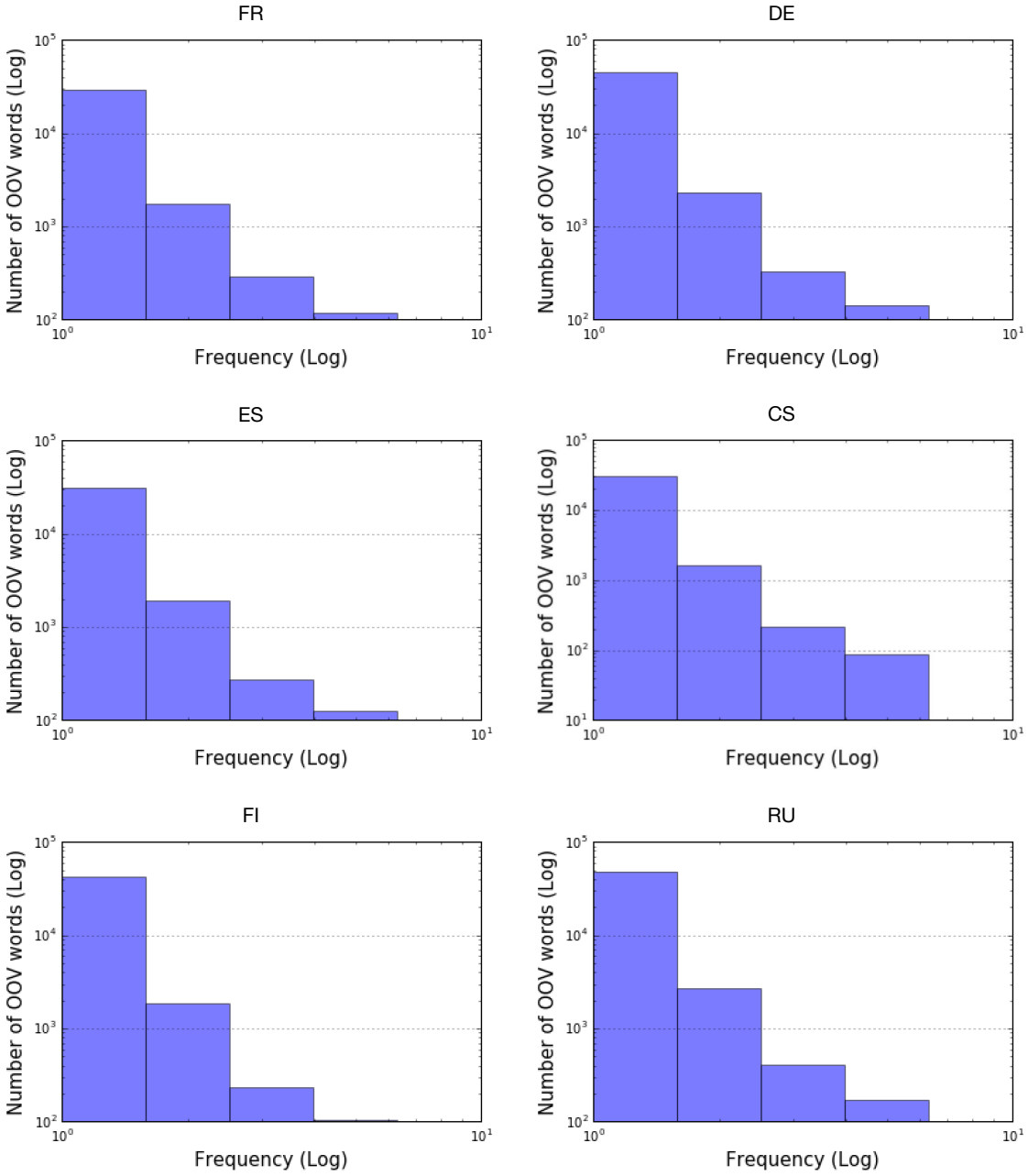

Figure 3.3: Histogram of OOV word frequencies in MWC Corpus in different languages.

Model Configuration For HCLM and HCLM with cache models, I used 600 dimensions for the character embeddings and the LSTMs have 600 hidden units for all the experiments. This keeps the model complexity to be approximately the same as previous works which used an LSTM with 1000 dimension. The baseline LSTM have 1000 dimensions for embeddings and reccurence weights.

For the cache model, I used cache size 100 in every experiment. All the parameters including character projection parameters are randomly sampled from uniform distribution from -0.08 to 0.08. The initial hidden and memory state of LSTM_{enc} and LSTM_{ctx} are initialized with zero. Mini-batches of size 25 are used for PTB experiments and 10 for WikiText-2, due to memory limitations. The sequences were truncated with 35 words. Then the words are decomposed to characters and fed into the model. A Dropout rate of 0.5 was used for all but the recurrent connections.

Learning The models were trained with the Adam update rule (Kingma and Ba, 2015) with a learning rate of 0.002. The maximum norm of the gradients was clipped at 10.

Evaluation I evaluated the models with bits-per-character (bpc) a standard evaluation metric for character-level language models. Following the definition in Graves (2013), bits-per-character is the average value of $-\log_2 p(w_t \mid \boldsymbol{w}_{<t})$ over the whole test set,

$$bpc = -\frac{1}{|\boldsymbol{c}|} \log_2 p(\boldsymbol{w}),$$

where $|\boldsymbol{c}|$ is the length of the corpus in characters.

3.5 Results

PTB , Tab. 3.4 summarizes results on the PTB dataset.[5] The baseline HCLM model achieved 1.276 bpc which is better performance than the LSTM with Zoneout regularization (Krueger et al., 2017). And HCLM with cache outperformed the baseline model with 1.247 bpc and achieved competitive results with state-of-the-art models with regularization on recurrence weights, which was not used in the experiments.

Expressed in terms of per-word perplexity (i.e., rather than normalizing by the length of the corpus in characters, I normalize by words and exponentiate), the test perplexity on

[5]Models designated with a * have more layers and more parameters.

HCLM with cache is 94.79. The performance of the unregularized 2-layer LSTM with 1000 hidden units on word-level PTB dataset is 114.5 and the same model with dropout achieved 87.0. Considering the fact that the character-level models are dealing with an open vocabulary without unknown tokens, the results are promising.

Method	Dev	Test
CW-RNN (Koutnik et al., 2014)	-	1.46
HF-MRNN (Mikolov et al., 2012)	-	1.41
MI-RNN (Wu et al., 2016)	-	1.39
ME n-gram (Mikolov et al., 2012)	-	1.37
RBN (Cooijmans et al., 2017)	1.281	1.32
Recurrent Dropout (Semeniuta et al., 2016)	1.338	1.301
Zoneout (Krueger et al., 2017)	1.362	1.297
HM-LSTM (Chung et al., 2017)	-	1.27
HyperNetwork (Ha et al., 2017)	1.296	1.265
LayerNorm HyperNetwork (Ha et al., 2017)	1.281	1.250
2-LayerNorm HyperLSTM (Ha et al., 2017)*	-	1.219
2-Layer with New Cell (Zoph and Le, 2017)*	-	1.214
LSTM	1.369	1.331
HCLM	1.308	1.276
HCLM with Cache	**1.266**	**1.247**

Table 3.4: Results on PTB Corpus (bits-per-character). HCLM augmented with a cache obtains the best results among models which have approximately the same numbers of parameter as single layer LSTM with 1,000 hidden units. * indicates models with more parameters.

WikiText-2 Tab. 3.5 summarizes results on the WikiText-2 dataset. The baseline, LSTM achieved 1.803 bpc and HCLM model achieved 1.670 bpc. The HCLM with cache outperformed the baseline models and achieved 1.500 bpc. The word level perplexity is 227.30, which is quite high compared to the reported word level baseline result 100.9 with LSTM with ZoneOut and Variational Dropout regularization (Merity et al., 2017). However, the character-level model is dealing with 76,136 types in training set and 5.87% OOV rate where the word level models only use 33,278 types without OOV in test set. The improvement

rate over the HCLM baseline is 10.2% which is much higher than the improvement rate obtained in the PTB experiment.

Method	Dev	Test
LSTM	1.758	1.803
HCLM	1.625	1.670
HCLM with Cache	**1.480**	**1.500**

Table 3.5: Results on WikiText-2 Corpus .

Multilingual Wikipedia Corpus (MWC) Tab. 3.6 summarizes results on the MWC dataset. Similarly to WikiText-2 experiments, LSTM is strong baseline. I observe that the cache mechanism improve performance in all languages. In English, HCLM with cache achieved 1.538 bpc where the baseline is 1.622 bpc. It is a 5.2% improvement. For other languages, the improvement rates were 2.7%, 3.2%, 3.7%, 2.5%, 4.7%, 2.7% in FR, DE, ES, CS, FI, RU respectively. The best improvement rate was obtained in Finnish.

	EN		FR		DE		ES		CS		FI		RU	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
LSTM	1.793	1.736	1.669	1.621	1.780	1.754	1.733	1.667	2.191	2.155	1.943	1.913	1.942	1.932
HCLM	1.683	1.622	1.553	1.508	1.666	1.641	1.617	1.555	2.070	2.035	1.832	1.796	1.832	1.810
HCLM with Cache	**1.591**	**1.538**	**1.499**	**1.467**	**1.605**	**1.588**	**1.548**	**1.498**	**2.010**	**1.984**	**1.754**	**1.711**	**1.777**	**1.761**

Table 3.6: Results on MWC Corpus (bits-per-character).

3.6 Analysis

In this section, I analyse the behavior of proposed model qualitatively. To analyse the model, I compute the following posterior probability which tells whether the model used the cache given a word and its preceding context. Let z_t be a random variable that says whether to use the cache or the LM to generate the word at time t. I would like to know, given the text $\boldsymbol{w}$, whether the cache was used at time t. This can be computed as follows:

$$
\begin{aligned}
p(z_t \mid \boldsymbol{w}) &= \frac{p(z_t, w_t \mid \mathbf{h}_t, \text{cache}_t)}{p(w_t \mid \mathbf{h}_t, \text{cache}_t)} \\
&= \frac{(1-\lambda_t) p_{ptr}(w_t \mid \mathbf{h}_t, \text{cache}_t)}{p(w_t \mid \mathbf{h}_t, \text{cache}_t)},
\end{aligned}
$$

where cache_t is the state of the cache at time t. I report the average posterior probability of cache generation excluding the first occurrence of w, $\overline{p(z \mid w)}$.

Word	$\overline{p(z \mid \boldsymbol{w})} \downarrow$	Word	$\overline{p(z \mid w)} \uparrow$
.	0.997	300	0.000
Lesnar	0.991	act	0.001
the	0.988	however	0.002
NY	0.985	770	0.003
Gore	0.977	put	0.003
Bintulu	0.976	sounds	0.004
Nerva	0.976	instead	0.005
,	0.974	440	0.005
UB	0.972	similar	0.006
Nero	0.967	27	0.009
Osbert	0.967	help	0.009
Kershaw	0.962	few	0.010
Manila	0.962	110	0.010
Boulter	0.958	Jersey	0.011
Stevens	0.956	even	0.011
Rifenburg	0.952	y	0.012
Arjona	0.952	though	0.012
of	0.945	becoming	0.013
31B	0.941	An	0.013
Olympics	0.941	unable	0.014

Table 3.7: Word types with the highest/lowest average posterior probability of having been copied from the cache while generating the test set. The probability tells whether the model used the cache given a word and its context. **Left:** Cache is used for frequent words (*the*, *of*) and proper nouns (*Lesnar*, *Gore*). **Right:** Character level generation is used for basic words and numbers.

Tab. 3.7 shows the words in the WikiText-2 test set that occur more than 1 time that are most/least likely to be generated from cache and character language model (words that occur only one time cannot be cache-generated). I see that the model uses the cache for proper nouns: *Lesnar*, *Gore*, etc., as well as very frequent words which are always stored somewhere in the cache such as single-token punctuation, *the*, and *of*. In contrast, the model

uses the language model to generate numbers (which tend not to be repeated): *300*, *770* and basic content words: *sounds*, *however*, *unable*, etc. This pattern is similar to the pattern found in empirical distribution of frequencies of rare words observed in prior wors (Church and Gale, 1995; Church, 2000), which suggests the model is learning to use the cache to account for bursts of rare words.

To look more closely at rare words, I also investigate how the model handles words that occurred between 2 and 100 times in the test set, but fewer than 5 times in the training set. Fig. 3.4 is a scatter plot of $\overline{p(z \mid w)}$ vs the empirical frequency in the test set. As expected, more frequently repeated words types are increasingly likely to be drawn from the cache, but less frequent words show a range of cache generation probabilities.

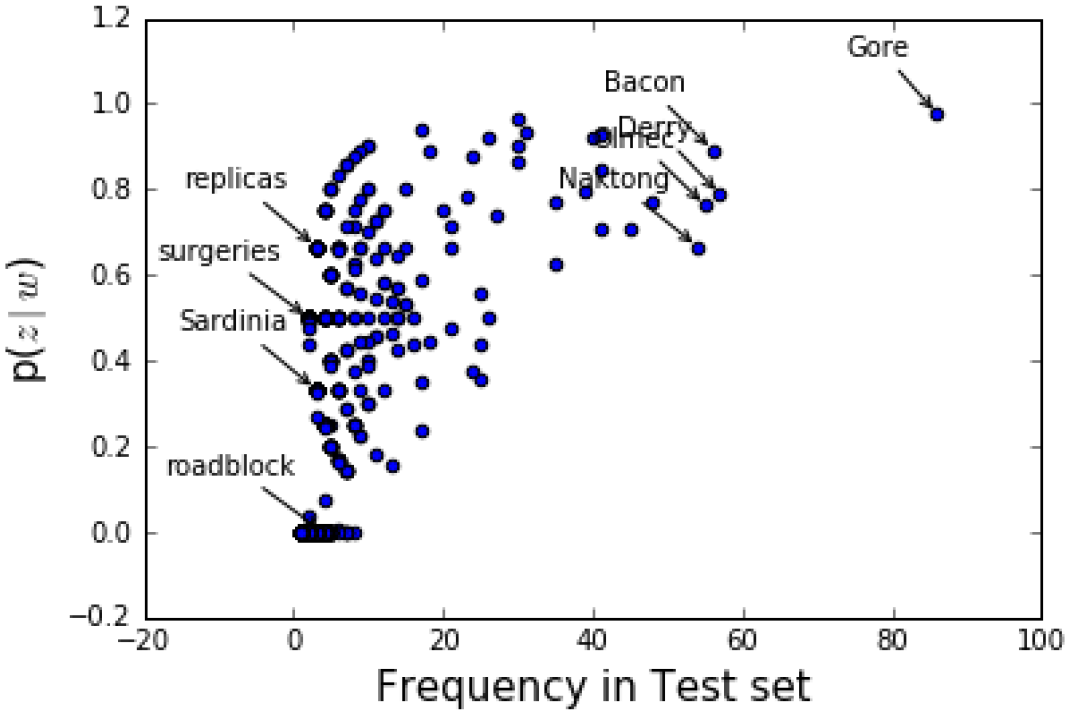

Figure 3.4: Average $p(z \mid w)$ of OOV words in test set vs. term frequency in the test set for words not obsered in the training set. The model prefers to copy frequently reused words from cache component, which tend to be names (upper right) while character level generation is used for infrequent open class words (bottom left).

Tab. 3.8 shows word types with the highest and lowest average $p(z \mid w)$ that occur fewer than 5 times in the training corpus. The pattern here is similar to the unfiltered list: proper

nouns are extremely likely to have been cache-generated, whereas numbers and generic (albeit infrequent) content words are less likely to have been.

Word	$\overline{p(z \mid \boldsymbol{w})} \downarrow$	Word	$\overline{p(z \mid w)} \uparrow$
Gore	0.977	770	0.003
Nero	0.967	246	0.037
Osbert	0.967	Lo	0.074
Kershaw	0.962	Pitcher	0.142
31B	0.941	Poets	0.143
Kirby	0.935	popes	0.143
CR	0.926	Yap	0.143
SM	0.924	Piso	0.143
impedance	0.923	consul	0.143
Blockbuster	0.900	heavyweight	0.143
Superfamily	0.900	cheeks	0.154
Amos	0.900	loser	0.164
Steiner	0.897	amphibian	0.167
Bacon	0.893	squads	0.167
filters	0.889	los	0.167
Lim	0.889	Keenan	0.167
Selfridge	0.875	sculptors	0.167
filter	0.875	Gen.	0.167
Lockport	0.867	Kipling	0.167
Germaniawerft	0.857	Tabasco	0.167

Table 3.8: Same as Table 7, except filtering for word types that occur fewer than 5 times in the training set. The cache component is used as expected even on rare words: proper nouns are extremely likely to have been cache-generated, whereas numbers and generic content words are less likely to have been; this indicates both the effectiveness of the prior at determining whether to use the cache and the burstiness of proper nouns.

3.7 Discussion

The results show that the HCLM outperforms a basic LSTM. With the addition of the caching mechanism, the HCLM becomes consistently more powerful than both the baseline

HCLM and the LSTM. This is true even on the PTB, which has no rare or OOV words in its test set (because of preprocessing), by caching repetitive common words such as *the*. In true open-vocabulary settings (i.e., WikiText-2 and MWC), the improvements are much more pronounced, as expected.

Computational complexity. In comparison with word-level models, the model has to read and generate each word character by character, and it also requires a softmax over the entire memory at every time step. However, the computation is still linear in terms of the length of the sequence, and the softmax over the memory cells and character vocabulary are much smaller than word-level vocabulary. On the other hand, since the recurrent states are updated once per character (rather than per word) in the model, the distribution of operations is quite different. Depending on the hardware support for these operations (repeated updates of recurrent states vs. softmaxes), the model may be faster or slower. However, the model will have fewer parameters than a word-based model since most of the parameters in such models live in the word projection layers, and I use LSTMs in place of these.

Non-English languages. For non-English languages, the pattern is largely similar to English. This is not surprising since morphological processes may generate forms that are related to existing forms, but these still have slight variations. Thus, they must be generated by the language model component (rather than from the cache). Still, the cache demonstrates consistent value in these languages.

Finally, the analysis of the cache on English does show that it is being used to model word reuse, particularly of proper names, but also of frequent words. While empirical analysis of rare word distributions predicts that names would be reused, the fact that cache is used to model frequent words suggests that effective models of language should have a means to generate common words as units. Finally, the model disfavors copying numbers from the cache, even when they are available. This suggests that it has learnt that numbers are not generally repeated (in contrast to names).

3.8 Related Work

Caching language models were proposed to account for burstiness by Kuhn and De Mori (1990), and recently, this idea has been incorporated to augment neural language models with a caching mechanism (Merity et al., 2017; Grave et al., 2017).

Open vocabulary neural language models have been widely explored (Sutskever et al., 2011; Mikolov et al., 2012; Graves, 2013, *inter alia*). Attempts to make them more aware of word-level dynamics, using models similar to the hierarchical formulation, have also been proposed (Chung et al., 2017).

The only models that are open vocabulary language modelling together with a caching mechanism are the nonparametric Bayesian language models based on hierarchical Pitman–Yor processes which generate a lexicon of word types using a character model, and then generate a text using these (Teh, 2006; Goldwater et al., 2009; Chahuneau et al., 2013b). These, however, do not use distributed representations on RNNs to capture long-range dependencies.

3.9 Conclusion

In this section, I proposed an explicit model of word creation and reuse using character-level neural language model with an adaptive cache which selectively assign word probability from past history or character-level decoding. And I empirically show that the proposed model efficiently models the word sequences and achieved better perplexity in standard language modelling datasets. To further validate the performance of the model on different languages, I collected multilingual wikipedia corpus for 7 typologically diverse languages. I also show that the model performs better than character-level models by modelling burstiness of words in local context.

The model proposed in this section assumes the observation of word segmentation. Thus, the model is not directly applicable to languages, such as Chinese and Japanese, where word segments are not explicitly observable. I will investigate a model which can marginalise word segmentation as latent variables in the next chapter.

Chapter 4

Word Discovery and Grounding

4.1 Introduction

How infants discover words that make up their first language is a long-standing question in developmental psychology (Saffran et al., 1996). Machine learning has contributed much to this discussion by showing that predictive models of language are capable of inferring the existence of word boundaries solely based on statistical properties of the input (Elman, 1990; Brent and Cartwright, 1996; Goldwater et al., 2009). However, there are two serious limitations of current models of word learning in the context of the broader problem of language acquisition. First, language acquisition involves not only learning what words there are ("the lexicon"), but also how they fit together ("the grammar"). Unfortunately, the best language models, measured in terms of their ability to predict language (i.e., those which seem to acquire grammar best), segment quite poorly (Chung et al., 2017; Wang et al., 2017; Kádár et al., 2018), while the strongest models in terms of word segmentation (Goldwater et al., 2009; Berg-Kirkpatrick et al., 2010) do not adequately account for the long-range dependencies that are manifest in language and that are easily captured by recurrent neural networks (Mikolov et al., 2010). Second, word learning involves not only discovering what words exist and how they fit together grammatically, but also determining their non-linguistic referents, that is, their grounding. The work that has looked at modelling acquisition of grounded language from character sequences—usually

The material in this chapter was presented in Kawakami et al. (2019).

in the context of linking words to a visually experienced environment—has either explicitly avoided modelling word units (Gelderloos and Chrupała, 2016) or relied on high-level representations of visual context that overly simplify the richness and ambiguity of the visual signal (Johnson et al., 2010; Räsänen and Rasilo, 2015).

This chapter introduces a single model that discovers words, learns how they fit together (not just locally, but across a complete sentence), and grounds them in learned representations of naturalistic non-linguistic visual contexts. I argue that such a unified model is preferable to a pipeline model of language acquisition (e.g., a model where words are learned by one character-aware model, and then a full-sentence grammar is acquired by a second language model using the words predicted by the first). The preference for the unified model may be expressed in terms of basic notions of simplicity (I require one model rather than two), and in terms of the Continuity Hypothesis of Pinker (1984), which argues that I should assume, absent strong evidence to the contrary, that children have the same cognitive systems as adults, and differences are due to them having set their parameters differently/immaturely.

In §4.2 I introduce a neural model of sentences that explicitly discovers and models word-like units from completely unsegmented sequences of characters. Since it is a model of complete sentences (rather than just a word discovery model), and it can incorporate multimodal conditioning context (rather than just modelling language unconditionally), it avoids the two continuity problems identified above. The model operates by generating text as a sequence of segments, where each segment is generated either character-by-character from a sequence model or as a single draw from a lexical memory of multi-character units. The segmentation decisions and decisions about how to generate words are not observed in the training data and marginalized during learning using a dynamic programming algorithm (§4.3).

The model depends crucially on two components. The first is, as mentioned, a lexical memory. This lexicon stores pairs of a vector (key) and a string (value). The strings in the lexicon are contiguous sequences of characters encountered in the training data; and the vectors are randomly initialized and learned during training. The second component is a regularizer (§4.4) that prevents the model from overfitting to the training data by overusing the lexicon to account for the training data.[1]

[1]Since the lexical memory stores strings that appear in the training data, each sentence could, in principle, be generated as a single lexical unit, thus the model could fit the training data perfectly while generalizing poorly. The regularizer penalizes based on the expectation of the powered length of each segment, preventing

The evaluation (§4.6–§4.9) looks at both language modelling performance and the quality of the induced segmentations, in both unconditional (sequence-only) contexts and when conditioning on a related image. First, I look at the segmentations induced by the model. I find that these correspond closely to human intuitions about word segments, competitive with the best existing models for unsupervised word discovery. Importantly, these segments are obtained in models whose hyperparameters are tuned to optimize validation (held-out) likelihood, whereas tuning the hyperparameters of the benchmark models using held-out likelihood produces poor segmentations. Second, I confirm findings (Kawakami et al., 2017; Mielke and Eisner, 2018) that show that word segmentation information leads to better language models compared to pure character models. However, in contrast to previous work, I realize this performance improvement without having to observe the segment boundaries. Thus, the model may be applied straightforwardly to Chinese, where word boundaries are not part of the orthography.

Ablation studies demonstrate that both the lexicon and the regularizer are crucial for good performance, particularly in word segmentation—removing either or both significantly harms performance. In a final experiment, I learn to model language that describes images, and I find that conditioning on visual context improves segmentation performance in the model (compared to the performance when the model does not have access to the image). On the other hand, in a baseline model that predicts boundaries based on entropy spikes in a character-LSTM, making the image available to the model has no impact on the quality of the induced segments, demonstrating again the value of explicitly including a word lexicon in the language model.

4.2 Model

I now describe the segmental neural language model (SNLM). Refer to Figure 4.1 for an illustration. The SNLM generates a character sequence $\boldsymbol{x} = x_1, \ldots, x_n$, where each x_i is a character in a finite character set Σ. Each sequence $\boldsymbol{x}$ is the concatenation of a sequence of segments $\underline{\boldsymbol{s}} = \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{|\underline{\boldsymbol{s}}|}$ where $|\underline{\boldsymbol{s}}| \leq n$ measures the length of the sequence in segments and each segment $\boldsymbol{s}_i \in \Sigma^+$ is a sequence of characters, $s_{i,1}, \ldots, s_{i,|\boldsymbol{s}_i|}$. Intuitively, each $\boldsymbol{s}_i$ corresponds to one word. Let $\pi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i)$ represent the concatenation of the characters of

this degenerate solution from being optimal.

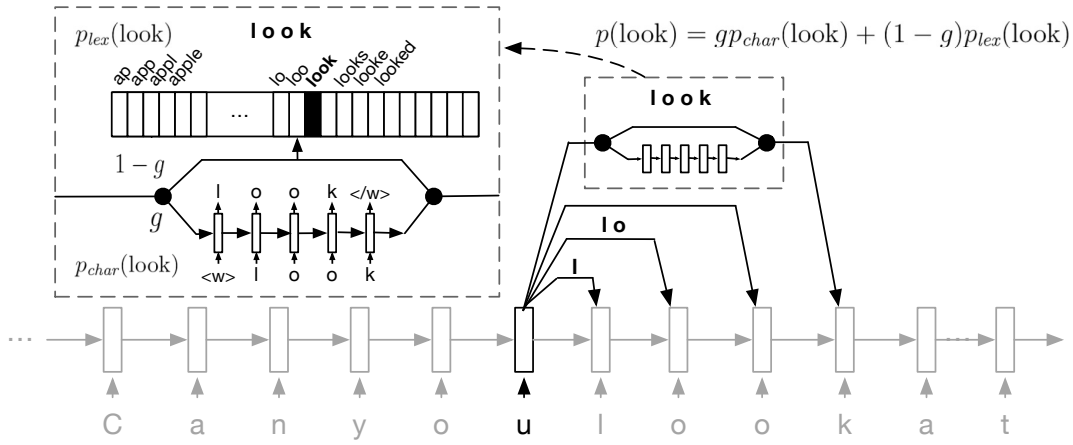

Figure 4.1: Fragment of the segmental neural language model while evaluating the marginal likelihood of a sequence. At the indicated time, the model has generated the sequence *Canyou*, and four possible continuations are shown.

the segments $\boldsymbol{s}_1$ to $\boldsymbol{s}_i$, discarding segmentation information; thus $\boldsymbol{x} = \pi(\underline{\boldsymbol{s}})$. For example if $\boldsymbol{x} = \texttt{anapple}$, the underlying segmentation might be $\underline{\boldsymbol{s}} = \texttt{an apple}$ (with $\boldsymbol{s}_1 = \texttt{an}$ and $\boldsymbol{s}_2 = \texttt{apple}$), or $\underline{\boldsymbol{s}} = \texttt{a nap ple}$, or any of the $2^{|\boldsymbol{x}|-1}$ segmentation possibilities for $\boldsymbol{x}$.

The SNLM defines the distribution over $\boldsymbol{x}$ as the marginal distribution over all segmentations that give rise to $\boldsymbol{x}$, i.e.,

$$p(\boldsymbol{x}) = \sum_{\underline{\boldsymbol{s}}:\pi(\underline{\boldsymbol{s}})=\boldsymbol{x}} p(\underline{\boldsymbol{s}}). \tag{4.1}$$

To define the probability of $p(\underline{\boldsymbol{s}})$, I use the chain rule, rewriting this in terms of a product of the series of conditional probabilities, $p(\boldsymbol{s}_t \mid \underline{\boldsymbol{s}}_{<t})$. The process stops when a special end-sequence segment $\langle/\text{S}\rangle$ is generated. To ensure that the summation in Eq. 4.1 is tractable, I assume the following:

$$p(\boldsymbol{s}_t \mid \underline{\boldsymbol{s}}_{<t}) \approx p(\boldsymbol{s}_t \mid \pi(\underline{\boldsymbol{s}}_{<t})) = p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t}), \tag{4.2}$$

which amounts to a conditional semi-Markov assumption—i.e., non-Markovian generation happens inside each segment, but the segment generation probability does not depend on memory of the previous segmentation decisions, only upon the sequence of characters $\pi(\underline{\boldsymbol{s}}_{<t})$ corresponding to the prefix character sequence $\boldsymbol{x}_{<t}$. This assumption has been employed in a number of related models to permit the use of LSTMs to represent rich history while

retaining the convenience of dynamic programming inference algorithms (Wang et al., 2017; Ling et al., 2017; Graves, 2012).

I model $p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t})$ as a mixture of two models, one that generates the segment using a sequence model and the other that generates multi-character sequences as a single event. Both are conditional on a common representation of the history, as is the mixture proportion.

Representing history To represent $\boldsymbol{x}_{<t}$, I use an LSTM encoder to read the sequence of characters, where each character type $\sigma \in \Sigma$ has a learned vector embedding $\mathbf{v}_\sigma$. Thus the history representation at time t is $\mathbf{h}_t = \mathrm{LSTM}_{enc}(\mathbf{v}_{x_1}, \ldots, \mathbf{v}_{x_t})$. This corresponds to the standard history representation for a character-level language model, although in general, I assume that the modelled data is not delimited by whitespace.

Character-by-character generation The first component model, $p_{char}(\boldsymbol{s}_t \mid \mathbf{h}_t)$, generates $\boldsymbol{s}_t$ by sampling a sequence of characters from an LSTM language model over Σ and a two extra special symbols, an end-of-word symbol $\langle/\mathrm{w}\rangle \notin \Sigma$ and the end-of-sequence symbol $\langle/\mathrm{s}\rangle$ discussed above. The initial state of the LSTM is a learned transformation of $\mathbf{h}_t$, the initial cell is $\mathbf{0}$, and different parameters than the history encoding LSTM are used. During generation, each letter that is sampled (i.e., each $s_{t,i}$) is fed back into the LSTM in the usual way and the probability of the character sequence decomposes according to the chain rule. The end-of-sequence symbol can never be generated in the initial position.

Lexical generation The second component model, $p_{lex}(\boldsymbol{s}_t \mid \mathbf{h}_t)$, samples full segments from lexical memory. Lexical memory is a key-value memory containing M entries, where each key, $\mathbf{k}_i$, a vector, is associated with a value $\boldsymbol{v}_i \in \Sigma^+$. The generation probability of $\boldsymbol{s}_t$ is defined as

$$\begin{aligned}
\mathbf{h}'_t &= \mathrm{MLP}(\mathbf{h}_t) \\
\mathbf{m} &= \mathrm{softmax}(\mathbf{K}\mathbf{h}'_t + \mathbf{b}) \\
p_{lex}(\boldsymbol{s}_t \mid \mathbf{h}_t) &= \sum_{i=1}^{M} m_i [\boldsymbol{v}_i = \boldsymbol{s}_t],
\end{aligned}$$

where $[\boldsymbol{v}_i = \boldsymbol{s}_t]$ is 1 if the ith value in memory is $\boldsymbol{s}_t$ and 0 otherwise, and $\mathbf{K}$ is a matrix obtained by stacking the $\mathbf{k}_i^\top$'s. This generation process assigns zero probability to most strings, but the alternate character model can generate all of Σ^+.

In this work, I fix the $\boldsymbol{v}_i$'s to be subsequences of at least length 2, and up to a maximum length L that are observed at least F times in the training data. These values are tuned as hyperparameters (See § 4.7 for details of the experiments).

Mixture proportion The mixture proportion, g_t, determines how likely the character generator is to be used at time t (the lexicon is used with probability $1 - g_t$). It is defined as $g_t = \sigma(\mathrm{MLP}(\mathbf{h}_t))$.

Total segment probability The total generation probability of $\boldsymbol{s}_t$ is thus

$$p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t}) = g_t p_{\textit{char}}(\boldsymbol{s}_t \mid \mathbf{h}_t) + (1 - g_t) p_{\textit{lex}}(\boldsymbol{s}_t \mid \mathbf{h}_t).$$

4.3 Inference

I am interested in two inference questions: first, given a sequence $\boldsymbol{x}$, evaluate its (log) marginal likelihood; second, given $\boldsymbol{x}$, find the most likely decomposition into segments $\underline{\boldsymbol{s}}^*$.

Marginal likelihood To efficiently compute the marginal likelihood, I use a variant of the forward algorithm for semi-Markov models (Yu, 2010), which incrementally computes a sequence of probabilities, α_i, where α_i is the marginal likelihood of generating $\boldsymbol{x}_{\leq i}$ and concluding a segment at time i. Although there are an exponential number of segmentations of $\boldsymbol{x}$, these values can be computed using $O(|\boldsymbol{x}|)$ space and $O(|\boldsymbol{x}|^2)$ time as:

$$\alpha_0 = 1, \qquad \alpha_t = \sum_{j=t-L}^{t-1} \alpha_j p(\boldsymbol{s} = \boldsymbol{x}_{j:t} \mid \boldsymbol{x}_{<j}). \tag{4.3}$$

By letting $x_{t+1} = \langle/\mathrm{S}\rangle$, then $p(\boldsymbol{x}) = \alpha_{t+1}$.

Most probable segmentation The most probable segmentation of a sequence $\boldsymbol{x}$ can be computed by replacing the summation with a $\max$ operator in Eq. 4.3 and maintaining backpointers.

4.4 Expected length regularization

When the lexical memory contains all the substrings in the training data, the model easily overfits by copying the longest continuation from the memory. To prevent overfitting, I

introduce a regularizer that penalizes based on the expectation of the exponentiated (by a hyperparameter β) length of each segment:

$$R(\boldsymbol{x}, \beta) = \sum_{\underline{\boldsymbol{s}}:\pi(\underline{\boldsymbol{s}})=\boldsymbol{x}} p(\underline{\boldsymbol{s}} \mid \boldsymbol{x}) \sum_{\boldsymbol{s}\in\underline{\boldsymbol{s}}} |\boldsymbol{s}|^{\beta}.$$

This can be understood as a regularizer based on the double exponential prior identified to be effective in previous work (Liang and Klein, 2009; Berg-Kirkpatrick et al., 2010). This expectation is a differentiable function of the model parameters. Because of the linearity of the penalty across segments, it can be computed efficiently using the above dynamic programming algorithm under the expectation semiring (Eisner, 2002). This is particularly efficient since the expectation semiring jointly computes the expectation and marginal likelihood in a single forward pass. For more details about computing gradients of expectations under distributions over structured objects with dynamic programs and semirings, see Li and Eisner (2009).

4.5 Training Objective

The model parameters are trained by minimizing the penalized log likelihood of a training corpus $\mathcal{D}$ of unsegmented sentences,

$$\mathcal{L} = \sum_{\boldsymbol{x}\in\mathcal{D}} [-\log p(\boldsymbol{x}) + \lambda R(\boldsymbol{x}, \beta)].$$

4.6 Datasets

I evaluate the model on both English and Chinese segmentation. For both languages, I used standard datasets for word segmentation and language modelling. I also use MS-COCO to evaluate how the model can leverage conditioning context information. For all datasets, I used train, validation and test splits.[2] Since the model assumes a closed character set, I removed validation and test samples which contain characters that do not appear in the training set. In the English corpora, whitespace characters are removed. In Chinese, they are not present to begin with. Table 4.1 summarizes dataset statistics.

[2]The data and splits used are available at
`https://s3.eu-west-2.amazonaws.com/k-kawakami/seg.zip`.

	Sentence			Char. Types			Word Types			Characters			Average Word Length		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
BR-text	7832	979	979	30	30	29	1237	473	475	129k	16k	16k	3.82	4.06	3.83
BR-phono	7832	978	978	51	51	50	1183	457	462	104k	13k	13k	2.86	2.97	2.83
PTB	42068	3370	3761	50	50	48	10000	6022	6049	5.1M	400k	450k	4.44	4.37	4.41
CTB	50734	349	345	160	76	76	60095	1769	1810	3.1M	18k	22k	4.84	5.07	5.14
PKU	17149	1841	1790	90	84	87	52539	13103	11665	2.6M	247k	241k	4.93	4.94	4.85
COCO	8000	2000	10000	50	42	48	4390	2260	5072	417k	104k	520k	4.00	3.99	3.99

Table 4.1: Summary of Dataset Statistics.

4.6.1 English

Brent Corpus The Brent corpus is a standard corpus used in statistical modelling of child language acquisition (Brent, 1999; Venkataraman, 2001).[3] The corpus contains transcriptions of utterances directed at 13- to 23-month-old children. The corpus has two variants: an orthographic one (**BR-text**) and a phonemic one (**BR-phono**), where each character corresponds to a single English phoneme. As the Brent corpus does not have a standard train and test split, and I want to tune the parameters by measuring the fit to held-out data, I used the first 80% of the utterances for training and the next 10% for validation and the rest for test.

English Penn Treebank (PTB) I use the commonly used version of the PTB prepared by Mikolov et al. (2010). However, since I removed space symbols from the corpus, the cross entropy results cannot be compared to those usually reported on this dataset.

4.6.2 Chinese

Since Chinese orthography does not mark spaces between words, there have been a number of efforts to annotate word boundaries. I evaluate against two corpora that have been manually segmented according different segmentation standards.

Beijing University Corpus (PKU) The Beijing University Corpus was one of the corpora used for the International Chinese Word Segmentation Bakeoff (Emerson, 2005).

Chinese Penn Treebank (CTB) I use the Penn Chinese Treebank Version 5.1 (Xue et al., 2005). It generally has a coarser segmentation than PKU (e.g., in CTB a full name, consisting of a given name and family name, is a single token), and it is a larger corpus.

[3] `https://childes.talkbank.org/derived`

4.6.3 Image Caption Dataset

To assess whether jointly learning about meanings of words from non-linguistic context affects segmentation performance, I use image and caption pairs from the COCO caption dataset Lin et al. (2014). I use 8000, 2000 and 10000 images for train, development and test set in order of integer ids specifying image in cocoapi[4] and use first annotation provided for each image. I will make pairs of image id and annotation id available from `https://s3.eu-west-2.amazonaws.com/k-kawakami/seg.zip`. The images are used to be conditional context to predict captions.

4.7 Experiments

I compare the model to benchmark Bayesian models, which are currently the best known unsupervised word discovery models, as well as to a simple deterministic segmentation criterion based on surprisal peaks (Elman, 1990) on language modelling and segmentation performance. Although the Bayeisan models are shown to able to discover plausible word-like units, I found that a set of hyperparameters that provides best performance with such model on language modelling does not produce good structures as reported in previous works. This is problematic since there is no objective criteria to find hyperparameters in fully unsupervised manner when the model is applied to completely unknown languages or domains. Thus, the experiments are designed to assess how well the models infers word segmentations of unsegmented inputs when they are trained and tuned to maximize the likelihood of the held-out text.

DP/HDP Benchmarks Among the most effective existing word segmentation models are those based on hierarchical Dirichlet process (HDP) models (Goldwater et al., 2009; Teh, 2006) and hierarchical Pitman–Yor processes (Mochihashi et al., 2009). As a representative of these, I use a simple bigram HDP model:

$$
\begin{aligned}
\theta_{\cdot} &\sim \mathrm{DP}(\alpha_0, p_0) \\
\theta_{\cdot|\boldsymbol{s}} &\sim \mathrm{DP}(\alpha_1, \theta_{\cdot}) \qquad \forall \boldsymbol{s} \in \Sigma^* \\
\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t &\sim \mathrm{Categorical}(\theta_{\cdot|\boldsymbol{s}_t}).
\end{aligned}
$$

[4] https://github.com/cocodataset/cocoapi

The base distribution, p_0, is defined over strings in $\Sigma^* \cup \{\langle/\text{S}\rangle\}$ by deciding with a specified probability to end the utterance, a geometric length model, and a uniform probability over Σ at a each position. Intuitively, it captures the preference for having short words in the lexicon. In addition to the HDP model, I also evaluate a simpler single Dirichlet process (DP) version of the model, in which the $\boldsymbol{s}_t$'s are generated directly as draws from $\text{Categorical}(\theta_\cdot)$. I use an empirical Bayesian approach to select hyperparameters based on the likelihood assigned by the inferred posterior to a held-out validation set.

By integrating out the draws from the DP's, it is possible to do inference using Gibbs sampling directly in the space of segmentation decisions. I use 1,000 iterations with annealing to find an approximation of the MAP segmentation and then use the corresponding posterior predictive distribution to estimate the held-out likelihood assigned by the model, marginalizing the segmentations using appropriate dynamic programs. The evaluated segmentation was the most probable segmentation according to the posterior predictive distribution.

In the original Bayesian segmentation work, the hyperparameters (i.e., α_0, α_1, and the components of p_0) were selected subjectively. To make comparison with the neural models fairer, I instead used an empirical approach and set them using the held-out likelihood of the validation set.

Deterministic Baselines Incremental word segmentation is inherently ambiguous (e.g., the letters *the* might be a single word, or they might be the beginning of the longer word *theater*). Nevertheless, several deterministic functions of prefixes have been proposed in the literature as strategies for discovering rudimentary word-like units hypothesized for being useful for bootstrapping the lexical acquisition process or for improving a model's predictive accuracy. These range from surprisal criteria (Elman, 1990) to sophisticated language models that switch between models that capture intra- and inter-word dynamics based on deterministic functions of prefixes of characters (Chung et al., 2017; Shen et al., 2018a).

In the experiments, I also include such deterministic segmentation results using (1) the surprisal criterion of Elman (1990) and (2) a two-level hierarchical multiscale LSTM (Chung et al., 2017), which has been shown to predict boundaries in whitespace-containing character sequences at positions corresponding to word boundaries. As with all experiments in this section, the BR-corpora for this experiment do not contain spaces.

SNLM Model configurations For each RNN based model I used 512 dimensions for the character embeddings and the LSTMs have 512 hidden units. All the parameters, including character projection parameters, are randomly sampled from uniform distribution from -0.08 to 0.08. The initial hidden and memory state of the LSTMs are initialized with zero. A dropout rate of 0.5 was used for all but the recurrent connections.

To restrict the size of memory, I stored substrings which appeared F-times in the training corpora and tuned F with grid search. The maximum length of subsequences L was tuned on the held-out likelihood using a grid search. Tab. 4.2 summarizes the parameters for each dataset. Note that I did not tune the hyperparameters on segmentation quality to ensure that the models are trained in a purely unsupervised manner assuming no reference segmentations are available.

	max len (L)	min freq (F)	λ
BR-text	10	10	7.5e-4
BR-phono	10	10	9.5e-4
PTB	10	100	5.0e-5
CTB	5	25	1.0e-2
PKU	5	25	9.0e-3
COCO	10	100	2.0e-4

Table 4.2: Hyperparameter values used.

For the image caption dataset, I extend the model with a standard attention mechanism in the backbone LSTM (LSTM_{enc}) to incorporate image context. For every character-input, the model calculates attentions over image features and use them to predict the next characters. As for image representations, I use features from the last convolution layer of a pre-trained VGG19 model Simonyan and Zisserman (2015).

4.8 Evaluation Metrics

Language modelling I evaluated the models with bits-per-character (bpc), a standard evaluation metric for character-level language models. Following the definition in Graves

(2013), bits-per-character is the average value of $-\log_2 p(x_t \mid \boldsymbol{x}_{<t})$ over the whole test set,

$$bpc = -\frac{1}{|\boldsymbol{x}|}\log_2 p(\boldsymbol{x}),$$

where $|\boldsymbol{x}|$ is the length of the corpus in characters. The bpc is reported on the test set.

4.9 Results

In this section, I first do a careful comparison of segmentation performance on the phonemic Brent corpus (BR-phono) across several different segmentation baselines, and I find that the model obtains competitive segmentation performance. Additionally, ablation experiments demonstrate that both lexical memory and the proposed expected length regularization are necessary for inferring good segmentations. I then show that also on other corpora, I likewise obtain segmentations better than baseline models. Finally, I also show that the model has superior performance, in terms of held-out perplexity, compared to a character-level LSTM language model. Thus, overall, the results show that I can obtain good segmentations on a variety of tasks, while still having very good language modelling performance.

Word Segmentation (BR-phono) Table 4.3 summarizes the segmentation results on the widely used BR-phono corpus, comparing it to a variety of baselines. **Unigram DP**, **Bigram HDP**, **LSTM suprisal** and **HMLSTM** refer to the benchmark models explained in §4.7. The ablated versions of the model show that without the lexicon ($-$memory), without the expected length penalty ($-$length), and without either, the model fails to discover good segmentations. Furthermore, I draw attention to the difference in the performance of the HDP and DP models when using subjective settings of the hyperparameters and the empirical settings (likelihood). Finally, the deterministic baselines are interesting in two ways. First, LSTM surprisal is a remarkably good heuristic for segmenting text (although I will see below that its performance is much less good on other datasets). Second, despite careful tuning, the HMLSTM of Chung et al. (2017) fails to discover good segments, although in their section they show that when spaces are present between, HMLSTMs learn to switch between their internal models in response to them.

Furthermore, the priors used in the DP/HDP models were tuned to maximize the likelihood assigned to the validation set by the inferred posterior predictive distribution, in contrast to previous papers which either set them subjectively or inferred them Johnson

and Goldwater (2009). For example, the DP and HDP model with subjective priors obtained 53.8 and 72.3 F1 scores, respectively (Goldwater et al., 2009). However, when the hyperparameters are set to maximize held-out likelihood, this drops obtained 56.1 and 56.9. Another result on this dataset is the feature unigram model of Berg-Kirkpatrick et al. (2010), which obtains an 88.0 F1 score with hand-crafted features and by selecting the regularization strength to optimize segmentation performance. Once the features are removed, the model achieved a 71.5 F1 score when it is tuned on segmentation performance and only 11.5 when it is tuned on held-out likelihood.

	P	R	F1
LSTM surprisal (Elman, 1990)	54.5	55.5	55.0
HMLSTM (Chung et al., 2017)	8.1	13.3	10.1
Unigram DP	63.3	50.4	56.1
Bigram HDP	53.0	61.4	56.9
SNLM ($-$memory, $-$length)	54.3	34.9	42.5
SNLM ($+$memory, $-$length)	52.4	36.8	43.3
SNLM ($-$memory, $+$length)	57.6	43.4	49.5
SNLM ($+$memory, $+$length)	**81.3**	**77.5**	**79.3**

Table 4.3: Summary of segmentation performance on phoneme version of the Brent Corpus (**BR-phono**).

Word Segmentation (other corpora) Table 4.4 summarizes results on the BR-text (orthographic Brent corpus) and Chinese corpora. As in the previous section, all the models were trained to maximize held-out likelihood. Here I observe a similar pattern, with the SNLM outperforming the baseline models, despite the tasks being quite different from each other and from the BR-phono task.

Word Segmentation Qualitative Analysis I show some representative examples of segmentations inferred by various models on the BR-text and PKU corpora in Table 4.5. As reported in Goldwater et al. (2009), I observe that the DP models tend to undersegment, keep long frequent sequences together (e.g., they failed to separate articles). HDPs do successfully prevent oversegmentation; however, I find that when trained to optimize held-out

		P	R	F1
BR-text	LSTM surprisal	36.4	49.0	41.7
	Unigram DP	64.9	55.7	60.0
	Bigram HDP	52.5	63.1	57.3
	SNLM	**68.7**	**78.9**	**73.5**
PTB	LSTM surprisal	27.3	36.5	31.2
	Unigram DP	51.0	49.1	50.0
	Bigram HDP	34.8	47.3	40.1
	SNLM	**54.1**	**60.1**	**56.9**
CTB	LSTM surprisal	41.6	25.6	31.7
	Unigram DP	61.8	49.6	55.0
	Bigram HDP	67.3	67.7	67.5
	SNLM	**78.1**	**81.5**	**79.8**
PKU	LSTM surprisal	38.1	23.0	28.7
	Unigram DP	60.2	48.2	53.6
	Bigram HDP	66.8	67.1	66.9
	SNLM	**75.0**	**71.2**	**73.1**

Table 4.4: Summary of segmentation performance on other corpora.

likelihood, they often insert unnecessary boundaries between words, such as *yo u*. The model's performance is better, but it likewise shows a tendency to oversegment. Interestingly, I can observe a tendency tends to put boundaries between morphemes in morphologically complex lexical items such as *dumpty 's*, and *go ing*. Since morphemes are the minimal units that carry meaning in language, this segmentation, while incorrect, is at least plasuible. Turning to the Chinese examples, I see that both baseline models fail to discover basic words such as 山间 (mountain) and 人们 (human).

Finally, I observe that none of the models successfully segment dates or numbers containing multiple digits (all oversegment). Since number types tend to be rare, they are usually not in the lexicon, meaning the model (and the H/DP baselines) must generate them as character sequences.

		Examples
BR-text	Reference	are you going to make him pretty this morning
	Unigram DP	areyou goingto makehim pretty this morning
	Bigram HDP	areyou go ingto make him p retty this mo rn ing
	SNLM	are you go ing to make him pretty this morning
	Reference	would you like to do humpty dumpty's button
	Unigram DP	wouldyoul iketo do humpty dumpty 's button
	Bigram HDP	would youlike to do humptyd umpty 's butt on
	SNLM	would you like to do humpty dumpty 's button
PKU	Reference	笑声 、 掌声 、 欢呼声 ， 在 山间 回荡 ， 勾 起 了 人们 对 往事 的 回忆 。
	Unigram DP	笑声 、 掌声 、 欢呼 声 ，在 山 间 回荡 ， 勾 起了 人们对 往事 的 回忆 。
	Bigram HDP	笑 声、 掌声 、 欢 呼声 ，在 山 间 回 荡， 勾 起了 人 们对 往事 的 回忆 。
	SNLM	笑声、 掌声 、 欢呼声 ， 在 山间 回荡 ， 勾起 了 人们 对 往事 的 回忆 。
	Reference	不得 在 江河 电缆 保护区 内 抛锚 、 拖锚 、 炸鱼 、 挖沙 。
	Unigram DP	不得 在 江河电缆 保护 区内抛锚、 拖锚 、炸鱼、挖沙 。
	Bigram HDP	不得 在 江 河 电缆 保护 区内 抛 锚、拖 锚 、 炸鱼、 挖沙 。
	SNLM	不得 在 江河 电缆 保护区 内 抛锚 、 拖锚、 炸鱼 、 挖沙 。

Table 4.5: Examples of predicted segmentations on English and Chinese.

Language modelling Performance The above results show that the SNLM infers good word segmentations. I now turn to the question of how well it predicts held-out data. Table 4.6 summarizes the results of the language modelling experiments. Again, I see that SNLM outperforms the Bayesian models and a character LSTM. Although there are numerous extensions to LSTMs to improve language modelling performance, LSTMs remain a strong baseline (Melis et al., 2018).

One might object that because of the lexicon, the SNLM has many more parameters than the character-level LSTM baseline model. However, unlike parameters in LSTM recurrence which are used every timestep, the memory parameters are accessed very sparsely. Furthermore, I observed that an LSTM with twice the hidden units did not improve the baseline with 512 hidden units on both phonemic and orthographic versions of Brent corpus but the lexicon could. This result suggests more hidden units are useful if the model does not have enough capacity to fit larger datasets, but that the memory structure adds other dynamics which are not captured by large recurrent networks.

Multimodal Word Segmentation Finally, I discuss results on word discovery with non-linguistic context (image). Although there is much evidence that neural networks can reliably

	BR-text	BR-phono	PTB	CTB	PKU
Unigram DP	2.33	2.93	2.25	6.16	6.88
Bigram HDP	1.96	2.55	1.80	5.40	6.42
LSTM	2.03	2.62	1.65	4.94	6.20
SNLM	**1.94**	**2.54**	**1.56**	**4.84**	**5.89**

Table 4.6: Test language modelling performance (bpc).

learn to exploit additional relevant context to improve language modelling performance (e.g. machine translation and image captioning), it is still unclear whether the conditioning context help to discover *structure* in the data. I turn to this question here. Table 4.7 summarizes language modelling and segmentation performance of the model and a baseline character-LSTM language model on the COCO image caption dataset. I use the Elman Entropy criterion to infer the segmentation points from the baseline LM, and the MAP segmentation under the model. Again, I find the model outperforms the baseline model in terms of both language modelling and word segmentation accuracy. Interestingly, I find while conditioning on image context leads to reductions in perplexity in both models, in the model the presence of the image further improves segmentation accuracy. This suggests that the model and its learning mechanism interact with the conditional context differently than the LSTM does.

To understand what kind of improvements in segmentation performance the image context leads to, I annotated the tokens in the references with part-of-speech (POS) tags and compared relative improvements on recall between SNLM ($-$image) and SNLM ($+$image) among the five POS tags which appear more than 10,000 times. I observed improvements on ADJ ($+4.5\%$), NOUN ($+4.1\%$), VERB ($+3.1\%$). The improvements on the categories ADP ($+0.5\%$) and DET ($+0.3\%$) are were more limited. The categories where I see the largest improvement in recall correspond to those that are likely *a priori* to correlate most reliably with observable features. Thus, this result is consistent with a hypothesis that the lexican is successfully acquiring knowledge about how words idiosyncratically link to visual features.

Segmentation State-of-the-Art The results reported are not the best-reported numbers on the English phoneme or Chinese segmentation tasks. As I discussed in the introduc-

	bpc↓	P ↑	R ↑	F1↑
Unigram DP	2.23	44.0	40.0	41.9
Bigram HDP	1.68	30.9	40.8	35.1
LSTM (−image)	1.55	31.3	38.2	34.4
SNLM (−image)	1.52	39.8	55.3	46.3
LSTM (+image)	1.42	31.7	39.1	35.0
SNLM (+image)	**1.38**	**46.4**	**62.0**	**53.1**

Table 4.7: Language modelling (bpc) and segmentation accuracy on COCO dataset. +image indicates that the model has access to image context.

tion, previous work has focused on segmentation in isolation from language modelling performance. Models that obtain better segmentations include the adaptor grammars (F1: 87.0) of Johnson and Goldwater (2009) and the feature-unigram model (88.0) of Berg-Kirkpatrick et al. (2010). While these results are better in terms of segmentation, they are weak language models (the feature unigram model is effectively a unigram word model; the adaptor grammar model is effectively phrasal unigram model; both are incapable of generalizing about substantially non-local dependencies). Additionally, the features and grammars used in prior work reflect certain English-specific design considerations (e.g., syllable structure in the case of adaptor grammars and phonotactic equivalence classes in the feature unigram model), which make them questionable models if the goal is to explore what models and biases enable word discovery in general. For Chinese, the best nonparametric models perform better at segmentation (Zhao and Kit, 2008; Mochihashi et al., 2009), but again they are weaker language models than neural models. The neural model of Sun and Deng (2018) is similar to the model without lexical memory or length regularization; it obtains 80.2 F1 on the PKU dataset; however, it uses gold segmentation data during training and hyperparameter selection, whereas the approach requires no gold standard segmentation data.

4.10 Related Work

Learning to discover and represent temporally extended structures in a sequence is a fundamental problem in many fields. For example in language processing, unsupervised learning of multiple levels of linguistic structures such as morphemes (Snyder and Barzilay, 2008), words (Goldwater et al., 2009; Mochihashi et al., 2009; Wang et al., 2014) and phrases (Klein and Manning, 2001) have been investigated. Recently, speech recognition has benefited from techniques that enable the discovery of subword units (Chan et al., 2017; Wang et al., 2017); however, in that work, the optimally discovered character sequences look quite unlike orthographic words. In fact, the model proposed by Wang et al. (2017) is essentially the model without a lexicon or the expected length regularization, i.e., ($-$memory, $-$length), which I have shown performs quite poorly in terms of segmentation accuracy. Finally, some prior work has also sought to discover lexical units directly from speech based on speech-internal statistical regularities (Kamper et al., 2016), as well as jointly with grounding (Chrupała et al., 2017).

4.11 Conclusion

In this chapter, I proposed a model that explicitly discover word boundaries at the same as learning predictive distributions over character sequences by parameterizing explicit latent variable model with implicit neural networks. While previous works which studied the word segmentation and language modeling in isolation has provided valuable insights (showing both what data is sufficient for word discovery with which models), this chapter shows that neural models offer the flexibility and performance to productively study the various facets of the problem in a more unified model. Moreover, I show that grounding to visual context further improves both word segmentation and language modeling.

While this work unifies several components that had previously been studied in isolation, the model assumes access to phonetic categories. The development of these categories likely interact with the development of the lexicon and acquisition of semantics (Feldman et al., 2013; Fourtassi and Dupoux, 2014), and thus subsequent work should seek to unify more aspects of the acquisition problem.

Chapter 5

Acoustic Modelling

5.1 Introduction

This chapter explores a method to learn phonetic structures from raw audio signals. Unlike previous chapters where the structures are in the text modality, the phonetic structures are in continuous signals. Moreover, the phonetic structures can be shared across different languages independently of language specific alphabets (§2.2). In this chapter, I focus on learning universal phonetic structures implicitly in continuous vector representations. The learned representations are evaluated in terms of the impact on speech recognition systems trained on the representations.

As reviewed in §2.1, the input representation of machine learning model strongly determines the difficulty faced by the learning algorithm, how much data the learner will require to find a good solution, and whether the learner generalizes out of sample and out of the domain of the training data. Representations (or features) that encode relevant information about data enable models to achieve good performance on downstream tasks, while representations that are invariant to factors that are not relevant to downstream tasks can further improve generalization. Traditionally, many invariances were hard-coded in feature extraction methods. For example, in image representations, geometric and photometric invariance has been investigated (Mundy et al., 1992; Van De Weijer et al., 2005). For acoustic representations, standard MFCC features are sensitive to additive noise and many

The material in this chapter was presented in Kawakami et al. (2020).

modifications have been proposed to overcome those limitations (Dev and Bansal, 2010; Kumar et al., 2011).

Recently, unsupervised representation learning algorithms have shown significant improvements at learning representations that correlate well with phonetic structure van den Oord et al. (2018); Kahn et al. (2020b) and improving downstream speech recognition performance Schneider et al. (2019); Baevski et al. (2019). Most of this work focused on learning representations from read English speech (from the LibriSpeech and LibriVox datasets) and evaluating the features when used to recognize speech in a rather similar domain (read English text). However, this approach to evaluation fails to test for the invariances that I would like good speech representations to have: robustness to domain shifts and transferability to other languages.

In the experiments I learn representations from 8000 hours of diverse and noisy speech, using an extended version of contrastive predictive coding model: bidirectional predictive models with dense residual connections (§5.2–§5.4), and evaluate the robustness and transferability of the representations by estimating how invariant they are to domain and language shifts. To do so, an ASR model is trained using the representations on one dataset but evaluated on the test sets of other datasets. In this experiment, I find that the representations derived from the large pretraining dataset lead the ASR model to be much more robust to domain shifts, compared to both log filterbank features as well as to pretraining just on LibriSpeech. I also train ASR models on 25 languages, including low-resource languages (e.g. Amharic, Fongbe, Swahili, Wolof), and show that the representations significantly outperform both standard features and those pretrained only on clean English data in the language transfer setup.

In summary, I confirm several increasingly common patterns that may be discerned in the literature on unsupervised representation learning, across a variety of modalities. First, scale matters: good representation learning requires a large amount of data. Second, unsupervised representations consistently improve robustness on downstream tasks. And finally, representations learned from multilingual data can transfer across many languages.

5.2 Contrastive Predictive Coding: CPC

Unsupervised representation learning methods rely on differentiable objectives which quantify the degree to which representations have succeeded at capturing the relevant characteristics in data. Mutual information measures relationships between random variables (Fano and Hawkins, 1961). Mutual information maximization techniques, that learn representations that describe data by maximizing mutual information between data and representation variables, have been explored for a long time in unsupervised representation learning (Linsker, 1988; Bell and Sejnowski, 1995). However, since the exact computation of mutual information is not tractable for continuous variables, recently many estimators have been proposed for enabling unsupervised representation learning with neural networks (Belghazi et al., 2018; van den Oord et al., 2018; Poole et al., 2019).

Contrastive predictive coding (van den Oord et al., 2018, CPC) is a mutual information maximization method that has been successfully applied to many modalities such as images and speech (Hénaff et al., 2020; Schneider et al., 2019). The objective is designed to extract features that allow the model to make long-term predictions about future observations. This is done by maximizing the mutual information of these features with those extracted from future timesteps. The intuition is that the representations capture different levels of structure dependent on how far ahead the model predicts. For example, if the model only predicts a few steps ahead, the resulting representations can capture local structures. On the other hand, if the model predicts further in the future, the representations will need to infer "slow features" (Wiskott and Sejnowski, 2002); more global structures such as phonemes, words and utterances in speech.

The overall unsupervised learning process is visualized in Figure 5.1. Given a raw audio signal of length L ($\boldsymbol{x} = x_1, x_2, \ldots, x_L$, $x_i \in \mathbb{R}$ where x_i represents the acoustic amplitude at time i), a function g_{enc} encodes the audio signals into vector representations ($\boldsymbol{z} = \boldsymbol{z}_1, \boldsymbol{z}_2 \ldots, \boldsymbol{z}_M$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$). Next, an autoregressive function g_{ar}, such as a recurrent neural network, summarizes the past representations and produces context vectors ($\boldsymbol{c} = \boldsymbol{c}_1, \boldsymbol{c}_2 \ldots, \boldsymbol{c}_M, \boldsymbol{c} \in \mathbb{R}^{d_c}$). The representations are learned to maximize mutual information between context vectors ($\boldsymbol{c}_t$) and future latent representations ($\boldsymbol{z} + \boldsymbol{k}$) as follows:

$$I(\boldsymbol{c}_t, \boldsymbol{z}_{t+k}) = \sum_{\boldsymbol{z}_{t+k}} p(\boldsymbol{c}_t, \boldsymbol{z}_{t+k} \mid k) \log \frac{p(\boldsymbol{z}_{t+k} \mid \boldsymbol{c}_t, k)}{p(\boldsymbol{z}_{t+k})}.$$

Since mutual information is not tractable for high dimensional data, it is common to use a lower-bound on the mutual information such as InfoNCE (van den Oord et al., 2018) which is a loss function based on noise contrastive estimation (Gutmann and Hyvärinen, 2010). Given a set $Z = \{\boldsymbol{z}_1, \dots \boldsymbol{z}_N\}$ which contains one positive sample from $p(\boldsymbol{z}_{t+k}|\boldsymbol{c}_t)$ and $N-1$ negative samples from a "noise" distribution $p(\boldsymbol{z})$, the approximated lower-bound is written as:

$$I(\boldsymbol{c}_t, \boldsymbol{z}_{t+k}) \geq \mathbb{E}_Z \left[\log \frac{f_k(\boldsymbol{c}_t, \boldsymbol{z}_{t+k})}{\frac{1}{N} \sum_{\tilde{\boldsymbol{z}} \in Z} f_k(\boldsymbol{c}_t, \tilde{\boldsymbol{z}})} \right] = \mathcal{L}_{tk}^{NCE},$$

where $f_k(\boldsymbol{c}_t, \boldsymbol{z}_{t+k})$ is a scoring function. I used the standard log-bilinear model as follows:

$$f_k(\boldsymbol{c}_t, \boldsymbol{z}_{t+k}) = \exp(\boldsymbol{c}_t^T \boldsymbol{W}_k \boldsymbol{z}_{t+k}).$$

The loss function I maximize is a sum of the InfoNCE loss for each step, $\mathcal{L}^{NCE} = \sum_t \sum_k \mathcal{L}_{tk}^{NCE}$ and the negatives are uniformly sampled from representations in the same audio signal ($\boldsymbol{z}$).

5.3 Methods

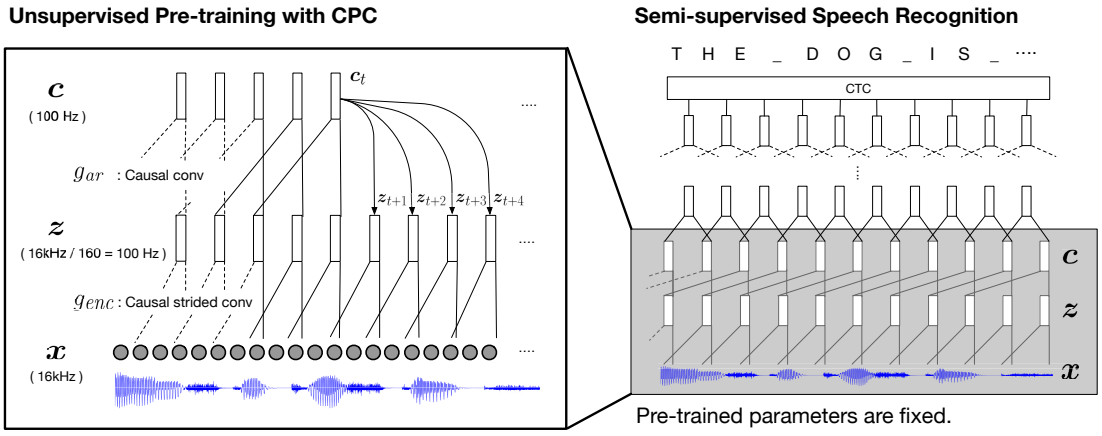

Figure 5.1: **Left**, unsupervised representation learning with forward contrastive predictive coding. The learned representations are fixed and used as inputs to a speech recognition model (**Right**).

In this section, I describe the models and objectives for unsupervised representation learning and downstream speech recognition. First, an acoustic feature extractor is trained

with a bidirectional variant of contrastive predictive coding on an unlabeled audio dataset. Next, the parameters of this model are frozen and its output representations are used as input to train various speech recognition models, potentially on a different or smaller labeled dataset (Figure 5.1).

5.3.1 Unsupervised learning with bi-directional CPC

Following the success of bidirectional models in representation learning (Peters et al., 2018; Devlin et al., 2019), I extend the original CPC method explained above with bidirectional context networks. The encoder function g_{enc} is shared for both directions, but there are two autoregressive models (g_{ar}^{fwd} and g_{ar}^{bwd}) which read encoded observations ($\boldsymbol{z}$) from the forward and backward contexts, respectively. The forward and backward context representations $\boldsymbol{c}_t^{fwd}, \boldsymbol{c}_t^{bwd}$ are learned with separate InfoNCE losses. When they are used for downstream tasks, a concatenation of two representations $\boldsymbol{c_t} = [\boldsymbol{c}_t^{fwd}; \boldsymbol{c}_t^{bwd}]$ is used. A similar technique has been used in image representation learning where representations are learned along different spatial dimensions (Hénaff et al., 2020).

All audio signals have a sampling rate of 16kHz and I normalize the mean and variance of the input signals over each utterance in order to mitigate volume differences between samples. For architectures, I use encoder and autoregressive models similar to Schneider et al. (2019). The encoder function g_{enc}, is a stack of causal convolutions with kernel sizes (10, 8, 4, 4, 4, 1, 1) and stride sizes (5, 4, 2, 2, 2, 1, 1), corresponding to a receptive field of 10 ms of audio. For autoregressive functions, I use a 13 layer causal convolution architecture with kernel sizes (1, 2, ..., 12, 13) and stride size 1, for both forward and backward functions. Layer-normalization across the temporal and feature dimensions is applied to every layer. Also, each layer has dense skip connections with layers below as in DenseNet (Huang et al., 2017). The objective function I optimize is the sum of the forward and backward InfoNCE losses (eq.5.2).

5.3.2 Semi-supervised speech recognition

Once the acoustic representations are trained, the resulting context vectors ($\boldsymbol{c}$) are used as inputs to character-level speech recognition models which predict transcriptions of audio-signals character by character. The model first predicts frame-level character probabilities with a series of convolution layers while the CTC forward algorithm (Graves et al., 2006)

calculates conditional probabilities of a transcription given an audio signal. The model parameters are trained to maximize the log likelihood of the data. The training terminates when the word error rate on the development set stops improving or the model has trained for more than a certain number of epochs. The models are evaluated on the standard word error rate (WER) metric on held-out test data. During training, the parameters in the speech recognition models are trained with supervision but the parameters of the representation models remain fixed. For decoding, I use greedy CTC decoding. In most experiments, I do not use a language model (LM) in order to isolate the effects of the acoustic representations, but I do include results with a 4-gram LM to facilitate comparisons with published results.

Common practice in unsupervised representation learning is to evaluate learned representations using a linear classifier rather than a more complex nonlinear model. However, I find that a simple linear layer followed by a CTC decoder does not have enough capacity to recognize speech. Thus, for the first set of experiments I use a smaller version of DeepSpeech2 (Amodei et al., 2016) to predict the frame-level character probabilities. The model has two 2d-convolutions with kernel sizes (11, 41) and (11, 21) and stride sizes (2, 2) and (1, 2) and one unidirectional recurrent neural network (GRU) on top of the output from the convolution layers. A linear transformation and a softmax function are applied to predict frame-level character probabilities. I refer to **DeepSpeech2 small** for the model specifics (Amodei et al., 2016). In order to further investigate how the representations interact with larger speech recognition models, I use the time-delay neural networks (**TDNN**) that are commonly used in speech recognition (Collobert et al., 2016; Kuchaiev et al., 2018). These consist of 17 layers of 1d-convolutions followed by 2 fully connected layers. Refer to OpenSeq2Seq for a detailed description.[1] These large models have been designed to perform well with log-filterbank features and purely supervised learning on large datasets, so they represent a challenging and informative test case for the value of learned representations.

5.4 Experiments and Results

5.4.1 Datasets

I collected publicly available speech datasets which cover a variety of types of speech (e.g. read and spoken), noise conditions and languages. For unsupervised pretraining I use a

[1] https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition/wave2letter.html

combination of datasets, using the audio but not any transcriptions, even when they are available. For semi-supervised learning (i.e., evaluation) on top of the representations I use the transcribed datasets following their standard train-test splits. Table 5.1 summarizes the datasets used for unsupervised learning and English speech recognition tasks.

Name	Language	Type	Hours
Audio Set	Multilingual	-	2500
AVSpeech	Multilingual	-	3100
Common Voice	Multilingual	read	430
LibriSpeech	English	read	960
WSJ	English	read	80
TIMIT	English	read	5
SSA	English	read	<1
Tedlium	English	spoken	440
Switchboard	English	spoken	310

Table 5.1: Summary of English Datasets.

Unlabeled speech pretraining corpus For pretraining, I collected a diverse and noisy speech corpus from several existing datasets: the subset of Audio Set (Gemmeke et al., 2017) containing speech examples, the audio part of AVSpeech (Ephrat et al., 2018), and the Common Voice (CV)[2] dataset in all 29 available languages. In addition I used the audio from TIMIT (Garofolo, 1993) and the Speech Accent Archive (Weinberger and Kunath, 2009), ignoring the transcriptions. Finally, I include the audio (again ignoring transcriptions) from the standard training splits of the evaluation datasets below. This collection spans a range of recording conditions, noise levels, speaking styles, and languages and amounts to about 8000 hours of audio.

Transcribed read English For evaluation, I look at the performance of the representations on a variety of standard English recognition tasks, as well as their ability to be trained on one and tested on another. For read English, I use LibriSpeech (Panayotov et al., 2015) and the Wall Street Journal (Paul and Baker, 1992).

[2] `https://voice.mozilla.org`

Transcribed spoken English To explore more extreme domain shifts, I additionally used conversational speech and public speaking datasets. I used Switchboard (Godfrey et al., 1992), a standard conversational speech recognition dataset consisting of two-sided telephone conversations (test only). Since the data was recorded more than 10 years ago and at a lower sampling rate than the other corpora, it presents a noisy and challenging recognition problem. Finally, I also use the Tedlium-3 (Hernandez et al., 2018) corpus, a large spoken English dataset containing 450 hours of speech extracted from TED conference talks. The recordings are clear, but there is some reverberation.

Transcription normalization Since I am comparing ASR systems trained on one dataset but evaluated on the test set of another, I normalize transcriptions to reduce systematic biases in the transfer condition. To do so, I use the format of the LibriSpeech dataset, which also ensures that the results are comparable with standard speech recognition systems on that task (Kuchaiev et al., 2018). For the other datasets, transcriptions are lowercased and unpronounced symbols (e.g., punctuation, silence markers) are removed. I also remove utterances containing numbers as they are transcribed inconsistently across and within datasets.

Transcribed multilingual speech In order to evaluate the transferability of the representations, I use speech recognition datasets in 4 African languages collected by the ALFFA project,[3] Amharic (Tachbelie et al., 2014), Fongbe (A. A Laleye et al., 2016), Swahili (Gelas et al., 2012), Wolof (Gauthier et al., 2016), for evaluation. These languages have unique phonological properties (e.g. height harmony) and phonetic inventories, making them a good contrast to English. These African languages are low-resource, each with 20 hours or less of transcribed speech. I also use 21 phonetically diverse languages from OpenSLR.[4] I only include (labeled) datasets from OpenSLR that containing more than 1GB of audio. When there is more than one dataset available for one language, I used the largest dataset. Table 5.2 summarizes the multilingual dataset statistics used in the evaluation.

5.4.2 Unsupervised Representation Learning

I train the model described above (§5.3.1) using the datasets described in the previous section (§5.4.1). Similarly to Schneider et al. (2019), audio signals are randomly cropped

[3] `http://alffa.imag.fr`
[4] `https://openslr.org`

with a window size 149,600 observations (9.35 seconds) and encoded with the model. The bidirectional contrastive predictive coding objective (Eq. 5.2) with prediction steps (k) 12 and negatives (N) 10 is optimized with the Adam optimizer with learning rate 0.0001. A batch size of 128 is used as well as a polynomial learning rate scheduler with power 2 and gradient clipping with maximum norm 5.0. Training was terminated at 4.2 million steps based on speech recognition performance on the dev (= validation) set of the LibriSpeech corpus.

5.4.3 Robustness

Robustness to shifts in domain, recording conditions, and noise levels is an important desideratum for a good ASR system, and I hypothesized that the diversity of the largest pretraining regime would improve robustness along these dimensions. In contrast, standard MFCC features have been tested in terms of noise robustness and it is known that such representations are sensitive to additive noise (Zhao and Wang, 2013). Moreover, speech recognition systems developed on top of such features are not robust when they are evaluated on out-of-domain datasets (Amodei et al., 2016).

To test whether the pretraining approach improves robustness, I evaluate speech recognition models trained on the learned representations on many different datasets so as to investigate benefit of using the representations learned from large-scale data. I compare ASR systems on all of the Wall Street Journal and LibriSpeech corpora with the same optimization as explained above and evaluate word error rate on different evaluation sets, such as phone call conversations (Switchboard).

Table 5.3 summarizes the results on models trained on Wall Street Journal, LibriSpeech or the Tedlium corpora and evaluated on different evaluation sets. **CPC-LibriSpeech** and **CPC-8k** indicate representations are learned from LibriSpeech and 8000h of speech datasets listed above respectively. The features trained on large-scale data consistently outperform other representations across different evaluation sets. The speech recognition models trained on the Wall Street Journal perform badly on phone call data in general. However, CPC representations learned on large datasets are more robust than those trained only on read English data (LibriSpeech).

5.4.4 Low-resource Languages

Thus far, all the experiments have compared the representations in terms of their impacts on English recognition tasks (although the pretraining dataset contains samples from many languages). I now turn to the question of whether these representations are suitable for driving recognition different languages with substantially different phonetic properties than English has. Specifically, I look at the performance on four languages—Amharic, Fongbe, Swahili, and Wolof—which manifest a variety of interesting phonological properties that are quite different from English. Evaluating on such languages provides insights into the phonetic space learned in the representations. Moreover, the non-English languages are low-resource in terms of speech recognition data, but have 2–20 million native speakers each. It is therefore valuable if the representations learned from large-scale unlabelled data can improve low-resource speech recognition. Although there is a chance that the large-scale pretraining dataset may contain some examples from those languages, I did not add any extra data specifically from those languages.

To test the cross-linguistic value of these features, I trained speech recognition models on low-resource languages (§5.4.1) and compare the relative reduction in WER by switching from standard spectrogram features and the learned representations. As these are very small datasets, I trained the same **DeepSpeech2 small** architecture with the Adam optimizer with a fixed learning rate of 0.0002 and gradient clipping with maximum norm 25.0 for all languages.

Figure 5.2 summarizes results. Again, I find that the CPC-8k representations outperform other features by a large margin and that the models trained on the representations trained on using the audio of (English-only) LibriSpeech do not perform even as well as basic spectrogram features. This suggests that the representations learned on large-scale data capture a phonetic space that generalizes across different languages, but that diversity of linguistic inputs is crucial for developing this universality.

5.4.5 Multilingual Transfer

As a final exploration of the transferability of the representations, I evaluate the representations on a diverse language set of languages with varying amounts of training data and compare the relative reductions in word error rate I obtain when using standard features and switching to the CPC-8k representations. As most of the dataset are small, I trained

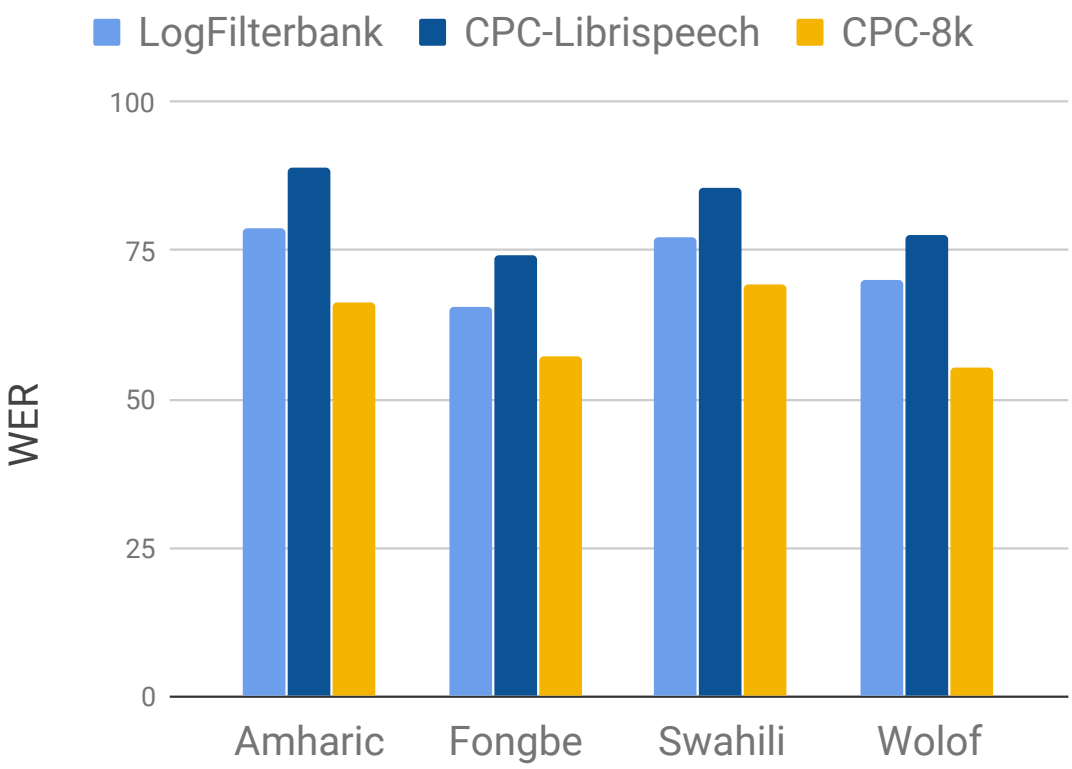

Figure 5.2: Speech recognition performance on low-resource African languages (in word error rate). CPC features trained on diverse datasets features significantly outperform baseline log-filterbank features whereas the features trained only on English underperform the baseline.

DeepSpeech2 small models with the Adam optimizer with a fixed learning rate of 0.0002 and applied gradient clipping with maximum norm 25.0, using the same configuration for all languages. Figure 5.3 summarizes results. Since the experiments above showed that CPC-LibriSpeech features performed badly, I only compare the relative error reduction with CPC-8k features over spectrogram features. In all cases, I find that the CPC-8k representations improve performance relative to spectorgram feature baselines. The largest improvement was obtained on Sundanese where the WER with spectrogram was 27.85 but dropped to 11.49 using CPC-8k features.

Discussion As the pre-training data did not have any language labels, it is unclear how many samples were seen for each language during pre-training. However, it is important to

know that the *uncurated* multilingual pre-training can improve speech recognition performance on many languages. These results suggests, in practice, that one could use a universal speech feature extractor for many languages instead of training one for each language individually Kannan et al. (2019).

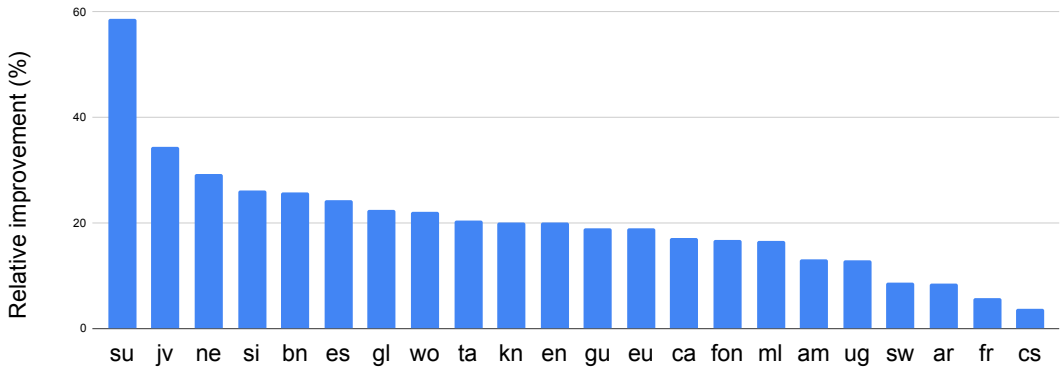

Figure 5.3: Relative improvements (in percentage) on speech recognition on many languages with CPC-8k features over Spectrogram features. Each column correspond to language code explained in Table 5.2. Note that **en** is Nigerian English and **fr** is African French.

5.4.6 Control: English Speech Recognition

Thus far, I have focused on robustness and transferability and seen that CPC-8k features offer considerable benefits in these dimensions compared to traditional features. It remains to demonstrate how well they work in powerful architectures where large amounts of labeled training data is available. To test this, I used 10% and 100% portions of LibriSpeech dataset to train speech recognition models, again comparing different features. The architecture is a standard **TDNN**. The speech recognition models are trained in the similar way as standard models (Collobert et al., 2016; Kuchaiev et al., 2018). The models are trained with Adam optimizer with learning rate 0.0002 and gradient clipping with a maximum norm 5.0 together with the polynomial learning rate decay method with power 2.0 is used over 200 epochs.[5]

Table 5.4 summarizes the results with **TDNN** models trained on different sizes of LibriSpeech dataset. I see that even if the speech recognition models have a large number

[5]These hyperparameters were chosen to give optimal performance with baseline log filterbank features, and used, unchanged for the learned features.

of parameters and are trained on plenty of supervised data, the learned representations still provide significant improvements. The pattern continues to hold if I use beam search decoding with a language model.[6] The **+ LM decoding** results are comparable to the OpenSeq2Seq benchmark, since I used the exact same LM and decoding algorithm as they used (Kuchaiev et al., 2018).

Although better results contain be obtained using newer architectures than TDNN (Park et al., 2019; Synnaeve et al., 2019), it still represents a standard and important recognition architecture and the results prove that the representations learned from diverse and noisy data can improve large speech recognition model on English in both low-data and high-data regimes.

5.5 Related Work

Unsupervised learning played an import role in the reintroduction of deep networks to speech processing (Hinton et al., 2012), as well as other application areas (Hinton et al., 2006; Bengio et al., 2007; Vincent et al., 2010). After a period of focusing on supervised techniques, unsupervised representation learning has recently seen a resurgence in a variety of modalities (Doersch and Zisserman, 2017; van den Oord et al., 2018; Donahue and Simonyan, 2019; Bachman et al., 2019) and has led to improved results, especially in low-data regimes (Hénaff et al., 2020; Schneider et al., 2019). In natural language processing, pretrained representations can outperform state-of-the-art system even in high data regimes (Mikolov et al., 2013; Devlin et al., 2019).

The last two years have produced a large amount of work on unsupervised speech representation learning. Some of this work has been evaluated in terms of its ability to perform phone recognition and similar audio classification tasks (van den Oord et al., 2018). Like us, Schneider et al. (2019); Baevski et al. (2019) applied learned representations to speech recognition tasks and evaluated on how well in-domain WER was improved. However, as I argued in the section, such an evaluation misses the opportunity to assess whether these systems become more robust to domain shift and to what extent the learned representations appropriate for different languages.

[6] `http://www.openslr.org/resources/11/4-gram.arpa.gz`

Finally, the ZeroSpeech challenges have explicitly looked at correlations between learned representations and phonetic structures that generalize across many languages and adapt to new speakers (Dunbar et al., 2017, 2019). Kahn et al. (2020b) learned representations with contrastive predictive coding on 60,000 hours of English speech and could show that their representations are correlated well with English phonetic structure; however, they did not evaluate these representations in a supervised speech recognizer.

Recently, there have been considerable improvements in purely supervised speech recognition systems. Data augmentation (Park et al., 2019), self-training (Synnaeve et al., 2019; Kahn et al., 2020a) have advanced the state-of-the-art performance on English speech recognition. It is likely that augmentation methods are orthogonal to the proposed improvements on universal speech representation learning, and that one could combine both to improve results even further. Additionally, the impact of data augmentation and self-training can be further assessed in terms of its impact on robustness using the methods proposed in this section.

5.6 Conclusion

I introduced an unsupervised speech representation learning method that implicitly discovers acoustic representations from up to 8000 hours of diverse and noisy speech data. I have shown, for the first time, that such pretrained representations lead speech recognition systems to be robust to domain shifts compared to standard acoustic representations, and compared to representations trained on smaller and more domain-narrow pretraining datasets. These representations were evaluated on a standard speech recognition setup where the models are trained and evaluated on in-domain data and also on transfer tasks where the models are evaluated on out-of-domain data. I obtained consistent improvements on 25 phonetically diverse languages including tonal and low-resource languages. This suggests I are making progress toward models that implicitly discover phonetic structure from large-scale unlabelled audio signals.

Language name	Code	Dataset	Hours
Amharic	am	ALFFA	18.3
Fongbe	fon	ALFFA	5.2
Swahilli	sw	ALFFA	8.9
Wolof	wo	ALFFA	16.8
Czech	cs	OpenSLR-6	15.0
Uyghur	ug	OpenSLR-22	20.2
Javanese	jv	OpenSLR-35	236.8
Sundanese	su	OpenSLR-36	265.9
Tunisian Arabic	ar	OpenSLR-46	4.5
Sinhala	si	OpenSLR-52	179.6
Bengali	bn	OpenSLR-53	172.3
Nepali	ne	OpenSLR-54	123.6
African French	fr	OpenSLR-57	13.7
Catalan	ca	OpenSLR-59	71.9
Malayalam	ml	OpenSLR-63	4.4
Tamil	ta	OpenSLR-65	5.7
Spanish	es	OpenSLR-67	19.6
Nigerian English	en	OpenSLR-70	39.5
Chilean Spanish	es	OpenSLR-71	5.7
Columbian Spanish	es	OpenSLR-72	6.1
Peruvian Spanish	es	OpenSLR-73	7.3
Basque	eu	OpenSLR-76	11.0
Galician	gl	OpenSLR-77	8.2
Gujarati	gu	OpenSLR-78	6.3
Kannada	kn	OpenSLR-79	6.7

Table 5.2: Summary of Multilingual Datasets.

	WSJ		LibriSpeech		Tedlium		Switchboard
	test92	test93	test-clean	test-other	dev	test	eval2000
WSJ							
LogFilterbank	16.78	23.26	46.27	73.27	58.61	62.55	96.44
CPC-LibriSpeech	11.89	15.66	31.05	56.31	45.42	47.79	83.08
CPC-8k	**10.77**	**14.99**	**29.18**	**51.29**	**38.46**	**39.54**	**69.13**
LibriSpeech							
LogFilterbank	14.42	21.08	6.43	20.16	26.9	25.94	61.56
CPC-LibriSpeech	14.28	20.74	6.91	21.6	26.53	27.14	63.69
CPC-8k	**13.31**	**18.88**	**6.25**	**19.10**	**21.56**	**21.77**	**53.02**
Tedlium							
LogFilterbank	20.35	27.23	24.05	47.27	18.75	19.31	74.55
CPC-LibriSpeech	15.01	19.52	17.77	36.7	15.28	15.87	61.94
CPC-8k	**13.17**	**17.75**	**16.03**	**32.35**	**13.67**	**13.88**	**47.69**

Table 5.3: Domain transfer experiments to test the robustness of the representations to domain shifts. The models are trained on the **Wall Street Journal**, **LibriSpeech** or **Tedlium** and evaluated on different evaluation sets. The results on in-domain evaluation sets are in gray color. All the results are without a language model.

	LibriSpeech							
	dev-clean		dev-other		test-clean		test-other	
	10%	100%	10%	100%	10%	100%	10%	100%
LibriSpeech								
LogFilterbank (OpenSeq2Seq)	-	6.67	-	18.67	-	6.58	-	19.61
LogFilterbank (ours)	19.83	6.63	38.97	18.77	19.65	6.43	41.26	20.16
CPC-LibriSpeech	15.07	6.70	33.55	19.77	14.96	6.91	36.05	21.60
CPC-8k	**13.92**	**6.20**	**30.85**	**17.93**	**13.69**	**6.25**	**32.81**	**19.10**
+ LM decoding								
LogFilterbank (OpenSeq2Seq)	-	4.75	-	13.87	-	4.94	-	15.06
LogFilterbank (ours)	12.49	4.87	28.71	14.14	12.29	5.04	31.03	15.25
CPC-LibriSpeech	9.66	4.87	24.72	14.34	9.41	5.05	26.77	16.06
CPC-8k	**8.86**	**4.35**	**22.10**	**12.96**	**8.70**	**4.72**	**24.15**	**14.47**

Table 5.4: Sample efficiency experiments with the **TDNN** trained and evaluated on **LibriSpeech**. The results are word error rate on the LibriSpeech development and evaluation sets. 10% vs. 100% indicates the amount of training data used. The section in **+ LM decoding** contain results with beamsearch decoding with a 4-gram language model. The underlined (OpenSeq2Seq) scores are taken from public benchmarks.

Chapter 6

Conclusion

This thesis started from the question of whether it is possible to design methods to incorporate linguistic structures while taking advantage of the representational power of distributed representations. In the previous chapters, I covered diverse linguistic structures and phenomena that are missing from recent works on statistical modelling of language such as word creation and reuse (§ 3), word discovery and grounding (§ 4), and universal phonetic structure (§ 5). The proposed methods are evaluated on new datasets and evaluation setups focusing on efficiency, generalization (robustness, multilinguality) and interpretability together with their performance on standard language modelling and speech recognition tasks. Overall, the results show the potential of combining explicit and implicit modelling of linguistic structures for better statistical models of language. I summarize the results and the findings and discuss future directions to conclude.

6.1 Explicit and Implicit modelling

This section summarizes how the balance between explicit and implicit modelling was explored in each section.

Word creation and reuse The open-vocabulary language model proposed in § 3 defines an explicit probabilistic model of character sequences that is a mixture of a word generation model and a lexical generation model. The model explicitly decides when to create a word character-by-character or retrieve from lexical memory that stores recently observed words. On the other hand, the morphological structures inside of words and syntactic structures between words are implicitly captured by hierarchical neural networks. The hierarchical

neural network is an implicit model but the model architecture itself provides inductive biases to let one model focus on morphological structures and the other focus on syntactic structures. The balance between explicit and implicit modelling enables us to use distributed representations and gradient-based learning. The results on language modelling show that the hierarchical neural networks is better than a completely implicit character-level LSTM in terms of held-out perplexity.

Word discovery and grounding The word discovery model extends the open-vocabulary language model with extra explicit latent variables that decide whether to put a word boundary after each character. Unlike the original model which avoids expensive combinatorial optimization, the proposed model needs to consider all possible segmentation decisions explicitly. In order to combine the explicit latent variable model with distributed representation and gradient base learning, we proposed to use a conditional semi-Markov model that enables us to compute the marginal efficiently with dynamic programming. The proposed method keeps the model differentiable at the same time as computing the marginal in tractable time. We found that alternative methods which approximate the marginal, such as REINFORCE and importance sampling, are not effective to discover meaningful structures.

We also proposed a regularization method (prior) that penalize the model based on the length of each segment. Although the length of each segment is not differentiable, we proposed to use the expected length of each segment that can be efficiently computed and differentiated together with the marginal likelihood. The experiments show that all the components are necessary to induce meaningful structure and to learn a good predictive distribution over character sequences.

Acoustic Modeling The representation learning models of text aim at representing different levels of structures from low-level morphological structures to high-level semantic structures in a single model. On the other hand, for acoustic modelling, I started from implicit learning of low-level phonetic structures. As the results show, the benefit of using implicit modelling is that the learned distributed representations are able to capture phonetic structures that generalize across different languages and they can be directly used to improve the robustness and efficiency of speech recognition models.

Explicit modelling of phonetic structures from raw audio is a promising direction to directly model how humans acquire language. However, from the machine learning point of view, it is hard to design an efficient learning algorithm on high-frequency audio data (the

standard 16kHz audio contains 16,000 data points in one second). In theory, the segmentation algorithm presented in § 4 should be applicable for discovering phonetic structures in audio signals, but in practice, it requires a lot of computing to explicitly marginalize all possible segmentations over the long sequences. Recently, discrete representation learning techniques, such as vector quatization (Oord et al., 2018) and variable-length encodings (Dieleman et al., 2021), can be used to reduce the sequence length while discovering meaningful units in continuous signals.

6.2 Efficiency, Generalization and Interpretability

As reviewed in §2.1, the quality of representations can be characterized by their efficiency, generalization and interpretability. In this thesis, I proposed datasets and evaluation frameworks to evaluate the representations on these aspects.

In § 3, the proposed language model was evaluated on the multilingual Wikipedia corpus (MWC) which consists of 7 different languages. These typologically diverse languages enable us to evaluate whether the proposed model is able to capture different use of morphology in each language. The corpus facilitated evaluating the generalization ability of language models on non-English languages in the follow-up works (Blevins and Zettlemoyer, 2019; Mielke and Eisner, 2019; Melis et al., 2019).

In § 4, I proposed a model that explicitly induces word-like units from unsegmented character sequences. Since the word segmentation task has been designed to simulate child language acquisition, it is standard to evaluate on small corpora of simple utterances directed at children. In addition to the small corpora, which evaluate how quickly the model capture regularities in the small amount of data, the models are evaluated on a standard language modelling dataset so that the model generalizes to complex sentences derived from news articles in English and Chinese. The interpretability of the learned structures is evaluated by comparing them against human-annotated references. Moreover, I propose a grounded word segmentation task that learns to discover and represent words from pairs of sentences and corresponding images. The new dataset is derived from the image caption dataset (MS-COCO). The dataset facilitates to study of multi-modal aspects of language learning in future works.

In § 5, I proposed new evaluation frameworks to test the robustness to domain shifts and multilingual generalization of the learned representations. Traditionally, ASR systems have been trained on each dataset (e.g. read speech, phone call etc). However, in the proposed framework, an ASR system trained on a single dataset is evaluated on diverse datasets in different domains. The experiments show that the ASR system trained on top of the learned representations works better on diverse datasets. The results suggest that the representations capture phonetic representations that generalize well to different domains. Moreover, the multilingual generalization is evaluated using speech recognition datasets in 25 different languages including low resource languages. The results open new research directions on universal speech representations.

6.3 Future Work

Statistical models of language have been studied for a long time as a model of human language acquisition and as a tool for practical applications such as spelling correction, speech recognition and translation. Recently, researchers found that large scale neural language models that learn implicit representations of language in more than 100 billion parameters are able to learn remarkable predictive distributions of language (Brown et al., 2020). The large-scale models trained on web-scale corpora are able to generate language much more fluently than previous models. However, it is still unclear whether such large-scale models are able to integrate the benefits of explicit modelling in terms of efficiency, generalization and interpretability. In this section, I propose several directions for future research.

Open vocabulary language modelling and lifelong learning are interesting setups that are closely aligned with the process of human language acquisition. The model needs to acquire words and phrases continuously from new linguistic observations as humans do. One approach is to directly apply the large-scale neural language models. However, as I presented in the thesis, implicit models may not be suitable for modelling quick adaptation to new context while being robust to the domain (topic) shifts over time. The memory mechanism presented in this thesis is a proposal to enable models to quickly adapt to the local context by reusing newly observed words and there are other promising alternatives proposed for closed-vocabulary language modelling such as retrieval memory (Guu et al., 2018; Khandelwal

et al., 2019). In addition to modelling, unlike standard language modelling evaluation where the training and test corpus are fixed, it is necessary to design new evaluation methods that evaluate efficient adaptation to new observations and robustness to domain shifts in a lifelong learning setup.

Another future direction is to further investigate the relationship between language learning and non-linguistic observations. In this thesis, I presented initial results that grounding to visual context improves the performance on language modelling and word discovery. In future work, it may be possible to learn from other modalities such as videos and sensory signals from robots. One limitation of the recent works on grounded language reviewed in § 2, and the experiment in this work, is that the models are trained and evaluated only on datasets designed for grounded language learning. Since the representations are specialized to learn the correspondence between text and other modalities, it is unclear whether the grounded linguistic representations are useful for standard languages processing tasks such as question answering and translation. Considering the fact that humans are able to learn language from different channels such as reading books (text only), watching the news (audio and video), and learning from each channel seem to help each other, it is interesting to implement such mechanisms in machine learning models.

Lastly, end-to-end language learning from speech, which learns different levels of linguistic representations from phonemes to semantics directly from speech signals, is a compelling extension of this thesis. As discussed in § 5, explicit modelling of high-frequency audio data is a challenging machine learning problem. However, as the evidence from cognitive science show (Saffran et al., 1996), humans are able to discover words from continuous speech signals before start using text. The efficient combination of explicit and implicit modelling may enable us to learn directly from audio and to integrate other linguistic phenomena explored in this thesis (e.g. word creation, reuse, grounding) into a single model.

References

Fréjus A. A Laleye, Laurent Besacier, Eugène C. Ezin, and Cina Motamed. First automatic fongbe continuous speech recognition system: Development of acoustic models and language models. In *Proc. FedCSIS*, 2016.

Hiyan Alshawi. *The core language engine*. MIT press, 1992.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proc. ICML*, 2016.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proc. CVPR*, 2018.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proc. NeurIPS*, 2019.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *Proc. ICLR*, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, 2015.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proc. LAW-VII and ID*, 2013.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. In *Proc. ICML*, 2018.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Proc. NeurIPS*, 2007.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Proc. NAACL*, 2010.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Terra Blevins and Luke Zettlemoyer. Better character language modeling through morphology. In *Proc. ACL*, 2019.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proc. ACL*, 2016.

Michael R Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1):71–105, 1999.

Michael R Brent and Timothy A Cartwright. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1):93–125, 1996.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Proc. NeurIPS*, 1993.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18 (4):467–480, 1992.

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

Victor Chahuneau, Eva Schlinger, Noah A Smith, and Chris Dyer. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687, 2013a.

Victor Chahuneau, Noah A. Smith, and Chris Dyer. Knowledge-rich morphological priors for bayesian language models. In *Proc. NAACL-HLT*, 2013b.

William Chan, Yu Zhang, Quoc Le, and Navdeep Jaitly. Latent sequence decompositions. In *Proc. ICLR*, 2017.

Eugene Charniak. *Statistical language learning*. MIT press, 1996.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. ICML*, 2020.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NeurIPS*, 2016.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. Learning to compose task-specific tree structures. In *Proc. AAAI*, 2018.

N. Chomsky. *Rules and Representations*. Columbia Classics in Philosophy. Columbia University Press, 1980. ISBN 9780231048279.

Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT press, 1965.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, 2005.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. In *Proc. ACL*, 2017.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *Proc. ICLR*, 2017.

Kenneth W Church. Empirical estimates of adaptation: the chance of two Noriegas is closer to $p/2$ than p^2. In *Proc. COLING*, 2000.

Kenneth W Church and William A Gale. Poisson mixtures. *Natural Language Engineering*, 1(2):163–190, 1995.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*, 2016.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. In *Proc. ICLR*, 2017.

Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. Translation using minimal recursion semantics. In *Proc. TMI*, 1995.

Ann A Copestake and Dan Flickinger. An open source grammar development environment and broad-coverage english grammar using hpsg. In *Proc. LREC*, 2000.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. Morphological smoothing and extrapolation of word embeddings. In *Proc. ACL*, 2016.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. *arXiv preprint cs/0205057*, 2002.

Carl De Marcken. The unsupervised acquisition of a lexicon from continuous speech. *arXiv preprint cmp-lg/9512002*, 1995.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Amita Dev and Poonam Bansal. Robust features for noisy speech recognition using mfcc computation from magnitude spectrum of higher order autocorrelation coefficients. *International Journal of Computer Applications*, 10(8):36–38, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, 2019.

Sander Dieleman, Charlie Nash, Jesse Engel, and Karen Simonyan. Variable-rate discrete representation learning. *arXiv preprint arXiv:2103.06089*, 2021.

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proc. ICCV*, pages 2051–2060, 2017.

Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Proc. NeurIPS*, 2019.

Mike Dowman. Addressing the learnability of verb subcategorizations with bayesian inference. In *Proceedings of the 22nd annual conference of the Cognitive Science Society*, pages 107–112. Cognitive Science Society Austin, TX, 2000.

Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux. The zero resource speech challenge 2017. In *Workshop Proc. ASRU*, 2017.

Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W Black, et al. The zero resource speech challenge 2019: Tts without t. In *Proc. INTERSPEECH*, 2019.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. In *Proc. NAACL*, 2016.

Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In *Proc. ACL*, 2002.

T Mark Ellison. The iterative learning of phonological constraints. *Computational Linguistics*, 20(3):1–32, 1994.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

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

Thomas Emerson. The second international Chinese word segmentation bakeoff. In *Workshop Proc. SIGHAN*, 2005.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4):1–11, 2018.

Robert M Fano and David Hawkins. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29:793–794, 1961.

Manaal Faruqui and Chris Dyer. Non-distributional word vector representations. In *Proc. ACL*, 2015.

Naomi H. Feldman, Thomas L. Griffiths, Sharon Goldwater, and James L. Morgan. A role for the developing lexicon in phonetic category acquisition. *Psychological Review*, 120 (4):751–778, 2013.

Abdellah Fourtassi and Emmanuel Dupoux. A rudimentary lexicon and semantics help bootstrap phoneme acquisition. In *Proc. EMNLP*, 2014.

Gottlob Frege. Über begriff und gegenstand. *Vierteljahrsschrift für wissenschaftliche Philosophie*, 16(2), 1892.

John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium*, 1993.

Elodie Gauthier, Laurent Besacier, Sylvie Voisin, Michael Melese, and Uriel Pascal Elingui. Collecting resources in sub-saharan african languages for automatic speech recognition: a case study of wolof. In *Proc. LREC*, 2016.

Hadrien Gelas, Laurent Besacier, and Francois Pellegrino. Developments of swahili resources for an automatic speech recognition system. In *Workshop Proc. SLTU*, 2012.

Lieke Gelderloos and Grzegorz Chrupała. From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning. In *Proc. COLING*, 2016.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI:721–741, 1984.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. ICASSP*, 2017.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proc. ICASSP*, 1992.

John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198, 2001.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, 2009.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Proc. NeurIPS*, 2014.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015.

Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *Proc. ICLR*, 2017.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, 2006.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. AISTATS*, 2010.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6: 437–450, 2018.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *Proc. ICLR*, 2017.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006.

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

Zellig S. Harris. From phoneme to morpheme. *Language*, 31(2):190–222, 1955. ISSN 00978507, 15350665. URL `http://www.jstor.org/stable/411036`.

Zellig S Harris. Recurrent dependence process: Morphemes by phoneme neighbors. *Mathematical Structures of Language*, 21:24–28, 1968.

Zellig S Harris. Morpheme boundaries within words: Report on a computer test. In *Papers in Structural and Transformational Linguistics*, pages 68–77. Springer, 1970.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

Harold Stanley Heaps. *Information retrieval: Computational and theoretical aspects*. Academic Press, Inc., 1978.

Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proc. ICML*, 2020.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *Proc. SPECOM*, 2018.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Geoffrey E Hinton. Distributed representations. 1984.

Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proc. ICLR*, 2019.

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

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. CVPR*, 2017.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Navdeep Jaitly and Geoffrey Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *Proc. ICASSP*, 2011.

Mark Johnson and Sharon Goldwater. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proc. NAACL*, pages 317–325, 2009.

Mark Johnson, Thomas L Griffiths, Sharon Goldwater, et al. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. *Proc. NeurIPS*, 19: 641, 2007.

Mark Johnson, Katherine Demuth, Michael Frank, and Bevan K. Jones. Synergies in learning words and their referents. In *Proc. NIPS*, 2010.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

Michael Irwin Jordan. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

Dan Jurafsky and James H Martin. *Speech & language processing*. 2020.

Ákos Kádár, Marc-Alexandre Côté, Grzegorz Chrupała, and Afra Alishahi. Revisiting the hierarchical multiscale LSTM. In *Proc. COLING*, 2018.

Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *Proc. INTERSPEECH*, 2020a.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. Libri-light: A benchmark for asr with limited or no supervision. In *Proc. ICASSP*, 2020b.

Herman Kamper, Aren Jansen, and Sharon Goldwater. Unsupervised word segmentation and lexicon induction discovery using acoustic word embeddings. *IEEE Transactions on Audio, Speech, and Language Processing*, 24(4):669–679, 2016.

Anjuli Kannan, Arindrima Datta, Tara N Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. Large-scale multilingual speech recognition with a streaming end-to-end model. In *Proc. INTERSPEECH*, 2019.

Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *Proc. CVPR*, 2015.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019.

Robert E Kass, Bradley P Carlin, Andrew Gelman, and Radford M Neal. Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to create and reuse words in open-vocabulary neural language modeling. In *Proc. ACL*, 2017.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to discover, ground and use words with segmental neural language models. In *Proc. ACL*, 2019.

Kazuya Kawakami, Luyu Wang, Chris Dyer, Phil Blunsom, and Aaron van den Oord. Learning robust and multilingual speech representations. In *Proc. EMNLP*, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *Proc. ICLR*, 2019.

Yoon Kim, Alexander M Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In *Proc. NAACL-HLT*, 2019.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Proc. NeurIPS*, 2015.

Dan Klein and Christopher D Manning. Distributional phrase structure induction. In *Workshop Proc. ACL*, 2001.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork RNN. In *Proc. ICML*, 2014.

David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, Aaron Courville, et al. Zoneout: Regularizing rnns by randomly preserving hidden activations. In *Proc. ICLR*, 2017.

Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Jason Li, Huyen Nguyen, Carl Case, and Paulius Micikevicius. Mixed-precision training for nlp and speech recognition with openseq2seq. *arXiv preprint arXiv:1805.10387*, 2018.

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 12(6): 570–583, 1990.

Kshitiz Kumar, Chanwoo Kim, and Richard M Stern. Delta-spectral cepstral coefficients for robust speech recognition. In *Proc. ICASSP*, 2011.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *Proc. ICLR*, 2016.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Zhifei Li and Jason Eisner. First-and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. EMNLP*, 2009.

Percy Liang and Dan Klein. Online EM for unsupervised models. In *Proc. NAACL*, 2009.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. Finding function in form: Compositional character models for open vocabulary word representation. In *Proc. EMNLP*, 2015.

Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Andrew Senior, Fumin Wang, and Phil Blunsom. Latent predictor networks for code generation. In *Proc. ACL*, 2017.

Ralph Linsker. An application of the principle of maximum information preservation to linear systems. In *Proc. NeurIPS*, 1988.

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

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *Proc. ICLR*, 2018.

Gábor Melis, Tomáš Kočiskỳ, and Phil Blunsom. Mogrifier lstm. In *Proc. ICLR*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proc. ICLR*, 2017.

Sebastian J. Mielke and Jason Eisner. Spell once, summon anywhere: A two-level open-vocabulary language model. In *Proc. NAACL*, 2018.

Sebastian J Mielke and Jason Eisner. Spell once, summon anywhere: A two-level open-vocabulary language model. In *Proc. AAAI*, 2019.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proc. INTERSPEECH*, 2010.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. Subword language modeling with neural networks. Technical report, Brno University of Technology, 2012.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*, 2013.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested Pitman–Yor language modeling. In *Proc. ACL*, 2009.

Joseph L Mundy, Andrew Zisserman, et al. *Geometric invariance in computer vision*, volume 92. MIT press Cambridge, MA, 1992.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proc. EMNLP*, 2015.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proc. EMNLP*, 2015.

Katerina Nicolaidis. Approval of new ipa sound: the labiodental flap. *Journal of the International Phonetic Association*, pages 261–261, 2005.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proc. LREC*, 2016.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proc. NeurIPS*, 2018.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Proc. ICASSP*, 2015.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *Proc. INTERSPEECH*, 2019.

Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proc. ACL*, 1992.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Steven Pinker. *Language learnability and language development*. Harvard University Press, 1984.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

Okko Räsänen and Heikki Rasilo. A joint model of word segmentation and meaning acquisition through cross-situational learning. *Psychological Review*, 122(4):792–829, 2015.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Jenny R Saffran, Richard N Aslin, and Elissa L Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Proc. AISTATS*, 2009.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In *Proc. INTERSPEECH*, 2019.

John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424, 1980.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. Recurrent dropout without memory loss. In *Proc. COLING*, 2016.

Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. In *Proc. ICLR*, 2018a.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proc. ICLR*, 2018b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

Burrhus F Skinner. Verbal behavior. *Language*, 11, 1957.

Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL*, 2008.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proc. EMNLP*, 2011.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proc. CIKM*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Mark Steedman. A very short introduction to ccg. *Preprint*, 1996.

Zhiqing Sun and Zhi-Hong Deng. Unsupervised neural word segmentation for Chinese via segmental language modeling. In *Proc. EMNLP*, 2018.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proc. ICML*, 2011.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end asr: from supervised to semi-supervised learning with modern architectures. In *Workshop Proc. ICML*, 2019.

Martha Tachbelie, Solomon Teferra Abate, and Laurent Besacier. Using different acoustic, lexical and language modeling units for asr of an under-resourced language - amharic. *Speech Communication*, 56, 2014.

Yee Whye Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. ACL*, 2006.

Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. Learning to interpret natural language commands through human-robot dialog. In *Proc. IJCAI*, 2015.

DS Touretzky and DW Wheeler. A connectionist implementation of cognitive phonology. Technical report, Carnegie Mellon University, 1989.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, 2015.

Alan M Turing and J Haugeland. *Computing machinery and intelligence*. MIT Press Cambridge, MA, 1950.

Joost Van De Weijer, Theo Gevers, and Arnold WM Smeulders. Robust photometric invariant features from the color tensor. *IEEE Transactions on Image Processing*, 15(1): 118–127, 2005.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proc. ICML*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017.

Anand Venkataraman. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):351–372, 2001.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11 (Dec):3371–3408, 2010.

Wilhelm von Humboldt. *Über die Verschiedenheit des menschlichen Sprachbaues: und ihren Einfluss auf die geistige Entwickelung des Menschengeschlechts*. Druckerei der Königlichen Akademie der Wissenschaften, 1836.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Workshop Proc. EMNLP*, 2019.

Chong Wang, Yining Wan, Po-Sen Huang, Abdelrahman Mohammad, Dengyong Zhou, and Li Deng. Sequence modeling via segmentations. In *Proc. ICML*, 2017.

Xiaolin Wang, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Empirical study of unsupervised Chinese word segmentation methods for SMT on large-scale corpora. In *Proc. ACL*, 2014.

Steven H Weinberger and Stephen Kunath. Towards a typology of english accents. *AACL Abstract Book*, 104, 2009.

Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

William A Woods. Semantics and quantification in natural language question answering. In *Advances in computers*, volume 17, pages 1–87. Elsevier, 1978.

Yuhuai Wu, Saizheng Zhang, Ying Zhang, Yoshua Bengio, and Ruslan R Salakhutdinov. On multiplicative integration with recurrent neural networks. In *Proc. NIPS*, 2016.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2): 207–238, 2005.

Zhilin Yang, Saizheng Zhang, Jack Urbanek, Will Feng, Alexander H Miller, Arthur Szlam, Douwe Kiela, and Jason Weston. Mastering the dungeon: Grounded language learning by mechanical turker descent. In *Proc. ICLR*, 2018.

Victor H Yngve, Elinor K Charney, and ES Klima. Mechanical translation. Technical report, Massachusetts Institute of Technology, 1962.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. In *Proc. ICLR*, 2016.

Shun-Zheng Yu. Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

Hai Zhao and Chunyu Kit. An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. In *Proc. IJCNLP*, 2008.

Xiaojia Zhao and DeLiang Wang. Analyzing noise robustness of mfcc and gfcc features in speaker identification. In *Proc. ICASSP*, 2013.

George K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *Proc. ICLR*, 2017.