Final Report

of the

Frederick Jelinek Memorial Summer Workshop

on

Neural Polysynthetic

Language Modelling

May 2020

Sixth Frederick Jelinek Memorial Summer Workshop

24 June – 02 August 2019

École de Technologie Supérieure
Montréal, Québec, Canada

Acknowledgements

Abstract

Many techniques in modern computational linguistics and natural language processing (NLP) make the assumption that approaches that work well on English and other widely used European (and sometimes Asian) languages are "language agnostic" – that is that they will also work across the typologically diverse languages of the world. In high-resource languages, especially those that are analytic rather than synthetic, a common approach is to treat morphologically-distinct variants of a common root (such as *dog* and *dogs*) as completely independent word types. Doing so relies on two main assumptions: that there exist a limited number of morphological inflections for any given root, and that most or all of those variants will appear in a large enough corpus (conditioned on assumptions about domain, etc.) so that the model can adequately learn statistics about each variant. Approaches like stemming, lemmatization, morphological analysis, subword segmentation, or other normalization techniques are frequently used when either of those assumptions are likely to be violated, particularly in the case of synthetic languages like Czech and Russian that have more inflectional morphology than English.

Within the NLP literature, agglutinative languages like Finnish and Turkish are commonly held up as extreme examples of morphological complexity that challenge common modelling assumptions. Yet, when considering all of the world's languages, Finnish and Turkish are closer to the average case in terms of synthesis. When we consider polysynthetic languages (those at the extreme of morphological complexity), even approaches like stemming, lemmatization, or subword modelling may not suffice. These languages have very high numbers of *hapax legomena* (words appearing only once in a corpus), underscoring the need for appropriate morphological handling of words, without which there is no hope for a model to capture enough statistical information about those words. Moreover, many of these languages have only very small text corpora, substantially magnifying these challenges.

To this end, we examine the current state-of-the-art in language modelling, machine translation, and predictive text completion in the context of four polysynthetic languages: Guaraní, St. Lawrence Island Yupik, Central Alaskan Yup'ik, and Inuktitut. We have a particular focus on Inuit-Yupik, a highly challenging family of endangered polysynthetic languages that ranges geographically from Greenland through northern Canada and Alaska to far eastern Russia. The languages in this family are extraordinarily challenging from a computational perspective, with pervasive use of derivational morphemes in addition to rich sets of inflectional suffixes and phonological challenges at morpheme boundaries. Finally, we propose a novel framework for language modelling that combines knowledge representations from finite-state morphological analyzers with Tensor Product Representations (Smolensky, 1990) in order to enable successful neural language models capable of handling the full linguistic variety of typologically variant languages.

Team Members

Team Leader

- Lane Schwartz
  Assistant Professor
  Department of Linguistics
  University of Illinois at Urbana-Champaign
  `lanes@illinois.edu`

  Lane Schwartz is an Assistant Professor of Computational Linguistics at the University of Illinois at Urbana-Champaign. His research centers on computational linguistics for endangered languages, with a focus on St. Lawrence Island Yupik; this includes work in polysynthetic language modelling, cognitively-motivated unsupervised grammar induction, and machine translation. He is one of the original developers of Joshua, an open source toolkit for tree-based statistical machine translation, and was a frequent contributor to Moses, the de-facto standard for phrase-based statistical machine translation.

Senior Members & Affiliates

- Francis Tyers
  Assistant Professor
  Department of Linguistics
  Indiana University
  `ftyers@iu.edu`

  Francis Tyers is an Assistant Professor of Computational Linguistics at Indiana University Bloomington. His research is focused on language technology for marginalized and indigenous languages and communities and he has worked extensively on the Uralic languages and the Turkic languages. In language technology his main interests are morphological modelling, using finite-state transducers and neural networks, dependency syntax and parsing, and machine translation. He is part of the core team of the Universal Dependencies project and secretary of the Apertium project — a free/open-source platform for machine translation.

- Lori Levin
  Research Professor
  Language Technologies Institute
  Carnegie Mellon University
  `levin@andrew.cmu.edu`

  Lori Levin is a Research Professor at the Language Technologies Institute at Carnegie Mellon University. She has 20 years experience in NLP for low-resource and endangered languages on several funded projects. She specializes in morphosyntax, language typology, and Construction Grammar.

- Christo Kirov
  Google
  `ckirov@gmail.com`

  Christo Kirov is a Research Software Engineer at Google, and was previously a Postdoctoral Fellow in the Center for Language and Speech Processing at Johns Hopkins University. His research has focused on computational morphophonology, especially in cross-linguistic, low-resource settings. He is one of the founders of the UniMorph project, which provides structured morphological paradigm data and related tools for many languages.

- Patrick Littell
  Research Officer
  National Research Council of Canada
  `patrick.littell@nrc-cnrc.gc.ca`

  Patrick Littell is a Research Officer in the Multilingual Text Processing team at the National Research Council of Canada (NRC-CNRC). His current research involves techniques for language technology development in very low-resource languages, by combining pre-trained multilingual models and knowledge-based rules and priors.

- Chi-kiu (Jackie) Lo
  Research Officer
  National Research Council of Canada
  `chikiu.lo@nrc-cnrc.gc.ca`

  Chi-kiu Lo is a Research Officer in the Multilingual Text Processing team at the National Research Council of Canada (NRC-CNRC). Her research interest is multilingual natural language processing with particular focuses on semantics in machine translation (MT), its quality evaluation and estimation. She designs a unified semantic-oriented MT quality evaluation and estimation metric, YiSi, that is readily available for evaluating translation quality in any language.

- Emily Prud'hommeaux
  Assistant Professor
  Department of Computer Science
  Boston College
  `prudhome@bc.edu`

  Emily Prud'hommeaux is an Assistant Professor of Computer Science at Boston College. Her research area is natural language and speech processing in low-resource settings, with a focus on developing tools to support the revitalization of the Haudenosaunee languages and other endangered languages of North America.

Graduate Students

- Hyunji Hayley Park
  Department of Linguistics
  University of Illinois at Urbana-Champaign
  hpark129@illinois.edu

  Hayley Park is a PhD student in Computational Linguistics at the University of Illinois at Urbana-Champaign. Her research focuses on computational linguistics and natural language processing for low-resource languages. Her recent projects include language modelling, grammar induction, morphological analysis and corpus digitization for low-resource languages.

- Kenneth Steimel
  Department of Linguistics
  Indiana University
  ksteimel@iu.edu

  Kenneth Steimel is a PhD candidate in Computational Linguistics at the University of Indiana Bloomington. His primary research interests are data-driven tagging of morphologically complex languages, particularly Bantu languages. His current research focuses on cross-language tagging for low resource Bantu languages.

- Rebecca Knowles
  Center for Language and Speech Processing, Johns Hopkins University &
  National Research Council of Canada
  Rebecca.Knowles@nrc-cnrc.gc.ca

  Rebecca Knowles is a Research Associate at the National Research Council of Canada (NRC-CNRC). She recently completed her Ph.D. in computer science at Johns Hopkins University. Her current research focuses on machine translation and computer aided translation.

- Jeffrey Micher
  Army Research Laboratory &
  Carnegie Mellon University
  jmicher@cs.cmu.edu

  Jeffrey Micher is a computer science researcher at Army Research Lab and a Ph.D. student at Carnegie Mellon Universtiy. His research interests include machine translation and morphological analysis of polysynthetic languages, specifically Inuktitut.

- Lonny Strunk
  Department of Linguistics
  University of Washington
  lonny.strunk@gmail.com

  Lonny Strunk is a Master's student in the computational linguistics program at the University of Washington. His research interests focus on language technology for indigenous languages. His current project is in the creation of a finite state morphological analyzer for his heritage language of Central Alaskan Yup'ik.

- Han Liu
  Department of Computer Science
  University of Colorado Boulder
  han.liu@colorado.edu

  Han Liu is a Ph.D. student in computer science at the University of Colorado Boulder. His research interests include natural language processing, human-centered machine learning, and human-AI collaboration.

Undergraduate Students

- Coleman Haley
  Johns Hopkins University
  `chaley7@jhu.edu`

  Coleman Haley is an undergraduate senior at Johns Hopkins University majoring in Computer Science and Cognitive Science. His research interests include neural interpretability in natural language processing, as well as NLP for morphologically and typologically diverse languages.

- Katherine J. Zhang
  Carnegie Mellon University
  `kjzhang@alumni.cmu.edu`

  Katherine Zhang is a member of the teaching staff at Carnegie Mellon University's Language Technologies Institute. She recently graduated from CMU with majors in Linguistics and Chinese Studies. Her research interests lie in corpus linguistics and Sino-Tibetan languages.

Graduate Student Affiliates

- Robbie Jimerson
  Rochester Institute of Technology
  `rcj2772@rit.edu`

  Robbie Jimerson is a Ph.D. candidate in Computing and Information Sciences at the Rochester Institute of Technology and a member of the Seneca Nation of Indians. His dissertation research focuses on developing robust language technologies to support the documentation and revitalization of the Seneca language and other endangered indigenous languages.

- Vasilisa Andriyanets
  Moscow Higher School of Economics
  `blindedbysunshine@gmail.com`

  Vasilisa Adriyanets is a recently graduated Masters student in computational linguistics from Higher School of Economics in Moscow, Russia. She has worked on computational approaches to processing a variety of languages, and specifically on morphological analysis for Russian and Chukchi.

Remote Student Affiliates

- Aldrian Obaja Muis, Naoki Otani, Jong Hyuk (Jay) Park, Zhisong Zhang
  Carnegie Mellon University
  `{amuis@cs, notani@cs, jp1@andrew, zhisongz@andrew}.cmu.edu`

Contents

Chapter 1

Introduction

Many techniques in modern computational linguistics and natural language processing (NLP) make the assumption that approaches that work well on English and other widely used European (and sometimes Asian) languages are "language agnostic" – that is that they will also work across the typologically diverse languages of the world.[1] In high-resource languages, especially those that are analytic rather than synthetic, a common approach is to treat morphologically-distinct variants of a common root (such as *dog* and *dogs*) as completely independent word types. Doing so relies on two main assumptions: that there exist a limited number of morphological inflections for any given root, and that most or all of those variants will appear in a large enough corpus (conditioned on assumptions about domain, etc.) so that the model can adequately learn statistics about each variant. Approaches like stemming, lemmatization, morphological analysis, subword segmentation, or other normalization techniques are frequently used when either of those assumptions are likely to be violated, particularly in the case of synthetic languages like Czech and Russian that have more inflectional morphology than English.

Within the NLP literature, agglutinative languages like Finnish and Turkish are commonly held up as extreme examples of morphological complexity that challenge common modelling assumptions. Yet, when considering all of the world's languages, Finnish and Turkish are closer to the average case in terms of synthesis. When we consider polysynthetic languages (those at the extreme of morphological complexity), approaches like stemming, lemmatization, or subword modelling may not suffice. These languages have very high numbers of *hapax legomena* (words appearing only once in a corpus), underscoring the need for appropriate morphological handling of words, without which there is no hope for a model to capture enough statistical information about those words. Moreover, many of these languages have only very small text corpora, substantially magnifying these challenges. The remainder of this work is structured as follows.

In Chapter 2 we briefly review the relevant background literature in finite-state morphology, language modelling, and machine translation. We review finite-state approaches to morphological analysis. We review the major approaches to language modelling, including n-gram language models, feed-forward language models, and recurrent neural language models.

In Chapter 3 we present a set of polysynthetic languages which we will consider throughout this work and detail the resources available for each. We have a particular focus on Inuit-Yupik, a highly challenging family of endangered polysynthetic languages that ranges geographically from Greenland through northern Canada and Alaska to far eastern Russia. The languages in this family are extraordinarily challenging from a computational perspective, with pervasive use of derivational morphemes in addition to rich sets of inflectional suffixes and phonological challenges at morpheme boundaries.

In Chapters 4–6 we examine the current state-of-the-art in language modelling, machine translation, and predictive text completion in the context of four polysynthetic languages: Guaraní, St. Lawrence Island Yupik, Central Alaskan Yup'ik, and Inuktitut. In Chapter 4 we present experiments and results on machine translation into, out of, and between polysynthetic languages; we carry out experiments between various Inuit-Yupik languages and English, as well as between Guaraní and Spanish, showing that multilingual approaches incorporating data from higher-resource members of the language family can effectively improve translation into lower-resource lan-

---

[1] Emily Bender provides a thorough discussion of this problem in `https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/`.
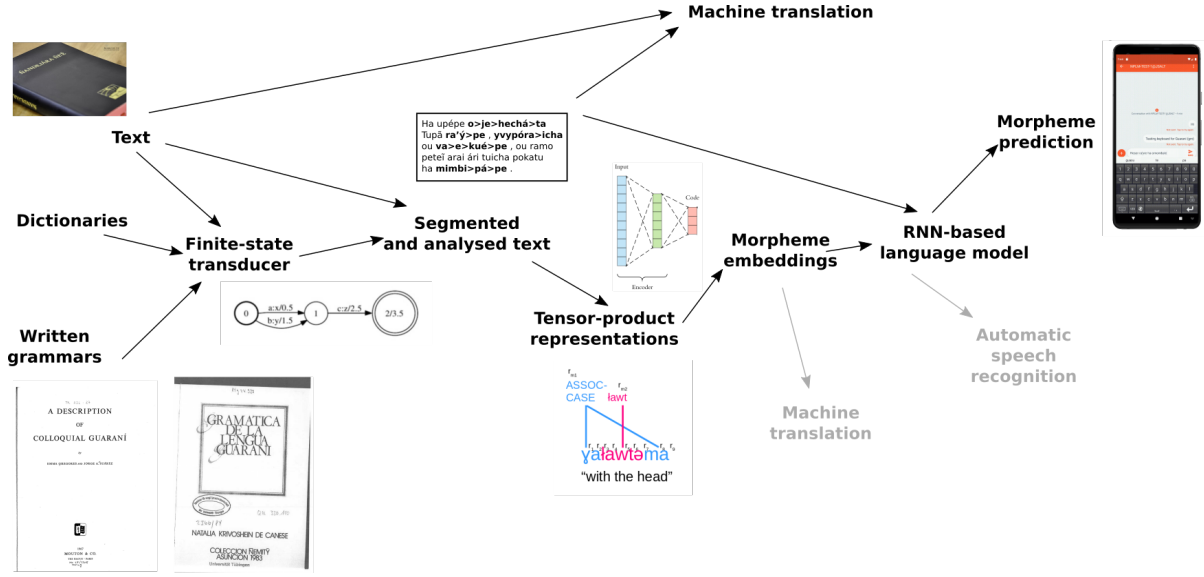
Figure 1.1: Overview of the tangible artefacts, models, and applications in this report. We start with all of the available resources for a given language, including (bi-)texts, grammars, and dictionaries. These are used to create finite-state morphological analyzers and MT systems (§4) directly. The finite-state morphological analyzers are then applied to corpora to create segmented or analyzed corpora (§2). These are used both to build language models (§5) and machine translation systems (§4) based on the segmented morphemes and to create interpretable morpheme-based language models using tensor product representations (§7). The final results are predictive keyboards that use morphemes as the unit of prediction (§6), with potential future work (greyed out) including automatic speech recognition and morpheme-based machine translation.

guages. In Chapter 5, we present language modelling experiments across a range of languages and vocabularies. In Chapter 6 we present practical applications which we anticipate will benefit from our language model and multilingual approaches, along with preliminary experimental results and discussion of future work.

Finally in Chapter 7 we present a core theoretical contribution of this work: a feature-rich open-vocabulary interpretable language model designed to support a wide range of typologically and morphologically diverse languages. This approach uses a novel neural architecture that explicitly model characters and morphemes in addition to words and sentences, making explicit use knowledge representations from finite-state morphological analyzers, in combination with Tensor Product Representations (Smolensky, 1990) to enable successful neural language models capable of handling the full linguistic variety of typologically variant languages. We present our conclusions in Chapter 8.

Chapter 2

Background

In this chapter we provide a brief overview of the background technologies that underlie this report, namely finite-state approaches to morphological analysis (§2.1), n-gram and neural language modelling techniques (§2.2), and neural machine translation (§2.3).

2.1 Finite-state morphology

Initial approaches to modelling the morphology of natural languages in the mid-20th century tended to focus on unidirectional algorithmic solutions to particular languages, implemented in general-purpose (rather than domain-specific) programming languages. These included generators, which generated wordforms from an analysis specification, analyzers, which returned possible analyses for a given word, and lemmatizers or stemmers which aimed to return a baseform, stem, or lemma given a wordform. These approaches had a number of downsides, the first being that the same code could not be used for analysis and generation, so for each language, separate code had to be written for these two tasks. In addition, descriptions could not be shared between related languages without much difficulty and there was little formalization.

In the early 1980s this changed with the introduction of finite-state morphology. In this formalization of morphology, the set of potential strings (wordform-analysis pairs) in a language is represented by a finite-state transducer. A finite-state transducer is a special class of finite-state automaton where each arc has both an input symbol and an output symbol. There are two main approaches to modelling morphophonological (or morphographemic) rules using finite-state approaches. The first consists of applying a sequence of rewrite rules in the form $\alpha \rightarrow \beta / \gamma \_ \delta$, where the alphabet symbol α is rewritten as β between γ and δ. The second approach is referred to as two-level morphology (Koskenniemi, 1983). In this approach, phonological rules are unordered constraints over possible symbol pairs. As Karttunen (1993) notes, the two approaches are formally equivalent and all phonological phenomena that can be described with one can be described with the other.

Given a description, a finite-state morphological analyzer can produce both analyses of surface tokens (e.g. sequences of tags and lemmas such as those found in interlinear glosses) and segmentations of surface tokens. Consider the output of the analyzer for the Guaraní sentence *Rehótapa che rendápe.* 'Will you come with me' in Example (1). The output includes the lemmas *ho* 'come', *che* 'my' and *tenda* 'place', person and number tags such as *<p2>* 'second person', *<sg>* 'singular', tags indicating word class, *<n>* 'noun' and *<v>* 'verb' among others.

(1) Input   Rehótapa che rendápe.
     Analysis   re<prn><p2><sg>+ho<v><iv>+ta<fti>+pa<qst>
                    che<prn><pos><p1><sg>
                    r<det>+tenda<n>+pe<post>
     Segmentation   Rehó>ta>pa che r>endá>pe

This is especially important for polysynthetic languages, as words can be made up of many morphemes, for example the word *ñaha'arõ'ỹetéva* 'that we did not expect at all' in the sentence *Oiko peteĩ mba'e ñaha'arõ'ỹetéva.*

"Something happened that we did not expect at all' can be decomposed as in Example (2) below.

(2)   Input   ñaha'arõ'ỹetéva
        Analysis   ña\<prn\>\<p1\>\<pl\>+ha'arõ+ỹ\<neg\>+ete\<emph\>+va\<subs\>
        Segmentation   ña\>ha'arõ\>'ỹ\>ete\>va

The amount of time required to develop a finite-state description can vary widely, but can be anywhere from two weeks, given a trained developer and a description of a related language — e.g. Kumyk in Washington et al. (2014) — to a year for a developer completely unfamiliar with the tools and language. The speed is also affected by the available resources such as grammatical descriptions and machine-readable lexicons.

One shortcoming of many finite-state morphological analyzers is an inability to assign probabilities to analyses. Table 2.1 depicts six example English sentences which each contain the word *wound*; each of these six uses is analyzed with a distinct linguistic analysis. When analyzing an English sentence that contains the word *wound*, an unweighted English morphological analyzer would posit all of these analyses, and would be unable to suggest which might be the most probable. Some finite-state morphological toolkits support the use of probabilities on

Analysis	Example	Frequency	Rel. frequency
'wind-PAST'	She wound the watch.	4	0.66
'wind-PP'	She had wound the watch.	1	0.16
'wound-N.SG'	The wound healed quickly.	1	0.16
'wound-INF'	Therefore I will wound you.	0	0
'wound-PRES'	They wound and they heal.	0	0
'wound-IMPER'	You wound me sir!	0	0

Table 2.1: List of analyses for the wordform *wound* in English, along with example sentences and frequency according to the English treebanks from the Universal Dependencies project (Nivre et al., 2016).

arcs in constructed finite-state transducers (Mohri, 2001). This means that it is possible to make analyzers and segmenters where the output is ranked, either by probability or by some other metric. Arc probability weights can be obtained from corpus statistics or from other measures. This is especially important for polysynthetic languages, where words may potential have many analyses. We describe the methods we used to weight our analyzers in Section 3.4.

2.2   Language modelling

A language model is any model that describes natural language. By that description, the finite-state models from the previous section could also be considered as a form of language model. In this section, however we use a narrower definition of language model as being a model of a probability distribution over a sequence of vocabulary items (characters, words).

Perhaps the simplest approximation to determine the probability of a sentence would be to use a unigram model over words. In such a model, the probability of a sentence is defined as the product of the probabilities of the individual words, which could be estimated by taking their relative frequency in a given corpus. While such a model could reasonably discriminate between the relative probabilities of sentences such as (a) "have a great trip" and (b) "have a superannuated tardigrade", it would not be able to distinguish the relative probability of (c) "great a have trip" and (a). A more accurate, but less tractable approximation would be to ask all speakers of a given language to rank all of the possible sentences in that language by some metric of 'goodness'. So the idea of language modelling is to find a tractable way to model the distribution of probability for sequences of linguistic symbols or tokens.

This simple model can be extended to n-gram language models (Shannon, 1948, 1951), whereby instead of modelling single units (characters, words), what is modelled is sequences of units. Thus in a bigram word model, the sequences modelled would be bigrams, e.g. {have a, a great, a trip} and {great a, a have, have trip} from examples (a) and (c) respectively. For languages where large amounts of monolingual training data are available,

language models of order 5–7 have been widely used in applications such as machine translation and automatic speech recognition.

However, as the model is extended to cover longer sequences, the problem of out-of-vocabulary (OOV) items becomes more severe. This happens when the sequence we are attempting to estimate the probability of does not appear in our model. This can be illustrated with the example in (b) above. The sequence "superannuated tardigrade" does not return any results with a search engine query on several major search engines. It is therefore highly likely that a bigram language model trained using all English text available on the internet would estimate the probability of this sequence to be zero, and therefore the probability of the entire sentence would also be zero. There are two techniques that have been developed to deal with this problem. Smoothing techniques reserve a small amount of the probability mass to distribute to unseen n-grams (Good, 1953; Jelineck and Mercer, 1980; Katz, 1987; Witten and Bell, 1991; Church and Gale, 1991; Ney et al., 1994; Kneser and Ney, 1995), while backoff techniques allow combinations of lower-order n-grams to be used to estimate the probability of higher-order ones. In example (b) the probabilities of 'superannuated' and 'tardigrade' would be used to estimate the probability of 'superannuated tardigrade'.

One of the issues with n-gram language models is that parameters are not shared between tokens and sequences. For example, the token 'wonderful' is as far from 'great' as is the token 'superannuated'. So if we have the sequence "have a wonderful trip", the other shared contexts that 'wonderful' and 'great' appear in are not taken into account. A way of dealing with this problem is to use distributional representations of individual tokens, as in Bengio et al. (2000, 2003). Here each token is represented by a vector of real numbers, embedding each token in a shared vector space. In these kind of language models it is still necessary to specify a fixed n-gram context, which means that the amount of context that can be taken into account is limited to a fixed-sized window for each token. Mikolov et al. (2010) describe using recurrent neural networks to model context to allow whole-sentence context to be taken into account. In addition they introduce efficient methods of training the distributional vectors such that corpora numbering in the billions of words can be used in training. In both the models proposed by Bengio et al. (2000) and Mikolov et al. (2010) each token is represented by a single vector. As evidenced from the examples above this is not always tenable, words in natural language are ambiguous (cf. *wound* and *trip* – 'to trip over something' or 'a nice trip'). In ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), each word vector is context dependent, both on external, sentence-level context, and on word-internal context, so even if a given token has not been seen before, the model can generalize from forms that have similar surface forms and appear in similar contexts. This would seem to be an ideal model for polysynthetic languages, however the downside is that these models typically contain very large numbers of parameters which in turn require very large amounts of training data, far more than is available for most endangered languages.

2.3 Machine Translation

In recent years, the machine translation community has gravitated toward neural approaches to machine translation. Midway through the 2010s, these began outperforming phrase-based statistical and other approaches in large-scale evaluations (Bojar et al., 2016). This success has driven a rapid sequence of approaches to building neural machine translation models, from sequence-to-sequence models (Sutskever et al., 2014), to models with attention (Bahdanau et al., 2015), to models that primarily rely on attention (Vaswani et al., 2017). In preparation for the workshop, we trained both statistical and neural machine translation models on the available training data. During the workshop, we focused solely on neural approaches to machine translation, and report those experiments in Chapter 4. As our experiments tended to examine variations of the input to the translation models rather than modifications to the networks themselves, we do not provide a thorough overview of the techniques here; for additional detail, please see the cited code and papers.

There does exist prior work on machine translation for polysynthetic languages, though it has generally been limited by small data sizes. In their recent overview of corpus resources for indigenous languages of the Americas, Mager et al. (2018a) note that most of the parallel corpora they found were quite small (less than 250,000 lines of text). Homola (2012) proposed the use of rule-based systems for polysynthetic languages, but this approach is still labor-intensive, as it requires the application of extensive linguistic knowledge or other tools. Monson et al. (2006) report on Mapudungun and Quechua to Spanish machine translation systems. Mager et al. (2018b) discuss challenges of translating between polysynthetic and fusional languages. This is not a complete account of all such work.

Of special note for the purposes of this work is existing research on two of the languages we worked on this summer: Inuktitut and Guaraní. For translation between Guaraní and Spanish, we are aware of an online gister (`http://iguarani.com/`) and Bible translations evaluated on stemmed output (Rudnick, 2018), and a system for translators called *Mainumby* by Gasser (2018). Previous work on translation between Inuktitut and English can be found in Micher (2018b), in which results of statistical machine translation for English and Inuktitut are reported. Micher makes use of a morphologically analyzed previous version of the Nunavut Hansard corpus to enhance SMT systems. Details on developing this corpus can be found in Micher (2018a). The FST-based analyzer (Farley, 2009) in combination with the neural analyzer (Micher, 2017) are used to morphologically analyze this data set. Klavans et al. (2018a) discuss some of the challenges of building such translation systems.

Chapter 3

Languages & Resources

A central issue that arises when conducting research on polysynthetic languages is the lack of resources: many polysynthetic languages are very low resource. Due to the need for corpora for use in language modelling efforts, an effort was directed towards locating existing corpora for polysynthetic languages and assessing their usability for different experiments. While we used only a subset of what we collected for experiments, this chapter provides an overview of all linguistic resources we gained access to in the process in order to offer a glimpse into available polysynthetic language resources.

In what follows, we provide short descriptions of the language families and languages involved and the corpora we collected. We briefly discuss the characteristics of polysynthetic languages based on descriptive statistics and the texts we selected for subsequent experiments. Details regarding corpus preprocessing are described in the context of experiments discussed in later chapters.

3.1 Language selection & data collection

We obtained corpora and resources for six languages: Chukchi, St. Lawrence Island Yupik, Central Alaskan Yup'ik, Inuktitut, Crow, and Guaraní. These languages were chosen from four different families, all of which are low-resource and polysynthetic. There was a focus in particular on the Inuit-Yupik-Unangan family, from which three of the languages were selected. The Inuit-Yupik-Unangan languages, historically known as Eskimo-Aleut, are a language family native to the Russian Far East, Alaska, Canada, and Greenland. The family is divided into two branches: Inuit-Yupik and Unangan. St. Lawrence Island Yupik, Central Alaskan Yup'ik, and Inuktitut belong to the Inuit-Yupik branch of the family.

In preparation for the workshop, we gathered spoken and written corpora for the selected polysynthetic languages. In addition to written and spoken corpora, where available, we also gathered dictionaries, reference grammars, and finite-state morphological analyzers. Table 3.1 provides a summary of the resources we had in each language. We refer to each language by name or by ISO 639-3 code.

Language	Code	Mono. text	Para. text	FST	Audio
Chukchi	ckt	✓		✓	✓
St. Lawrence Island Yupik	ess	✓	✓	✓	✓
Central Alaskan Yup'ik	esu	✓	✓		
Inuktitut	iku	✓	✓	✓	
Crow	cro				✓
Guaraní	grn	✓	✓	✓	

Table 3.1: Overview of languages and resources: monolingual text, parallel text, finite state transducers, and audio data.

3.1.1   Chukchi

Chukchi (`ckt`) is the most widely spoken language in the Chukotko-Kamchatkan family, with approximately 5000 speakers. The Chukotko-Kamchatkan languages are native to the Russian Far East, and Chukchi is spoken in the easternmost part, mainly on the Chukotka Peninsula.

  We obtained audio data and transcripts for Chukchi from `http://chuklang.ru`, a website dedicated to materials and research on Chukchi funded by the Russian Science Foundation. The audio data contains two books of the Bible, the Book of Jonah and the Gospel of Luke, and short stories in the language. The stories represent a valuable resource for the endangered language. The transcripts are in both Latin and Cyrillic scripts. There also exists a prototype finite-state morphological analyzer for Chukchi (Andriyanets and Tyers, 2018). This analyzer was expanded on during the workshop using the transcripts of the audio data.

3.1.2   St. Lawrence Island Yupik

St. Lawrence Island Yupik (`ess`) is an endangered language in the Inuit-Yupik family spoken on St. Lawrence Island, Alaska and on the Chukokta Peninsula of the Russian Far East. We collected a corpus consisting primarily of scanned and digitized books, including educational materials (Apassingok et al., 1993, 1994, 1995), oral narratives (Nagai, 2001; Apassingok et al., 1985, 1987, 1989; Slwooko, 1977, 1979) and a reference grammar (Jacobson, 2001). In addition, we made use of the Yupik translation of the New Testament[1] (Wycliffe, 2018). We made use of the Chen and Schwartz (2018) finite-state morphological analyzer, which was based on the Yupik grammar of Jacobson (2001) and incorporated Yupik lexical entries from the Badten et al. (2008) dictionary.

3.1.3   Central Alaskan Yup'ik

Central Alaskan Yup'ik (`esu`) is an official language of Alaska that is spoken by about 10,000 speakers in the western and southwestern parts of the state. There are five major dialects of Central Alaskan Yup'ik, of which General Central Yup'ik (Yugtun) is the most widely spoken.

  This workshop made use of a Yup'ik translation[2] of the Bible. As one of our team members speaks the language, we were able to align it with a corresponding English Bible (Good News Translation, Today's English Version, Second Edition). The parallel data were used for both machine translation and language modelling experiments. Additionally, the Yup'ik Bible and a dictionary (Jacobson, 1984) were used to begin development on a Yup'ik finite-state morphological analyzer.

3.1.4   Inuktitut

Inuktut (a term that includes the variants Inuktitut and Inuinnaqtun) is one of the official languages of Nunavut, the largest territory of Canada, and is spoken by approximately 39,770 people in Canada (Statistics Canada, 2017). It also has official recognition in several other areas and is part of the Inuit-Yupik-Unangan language family. Inuktut can be written in syllabics or in roman orthography, and regional variations use different special characters and spelling conventions.

  As Inuktut is an official language of government in Nunavut, there exist some resources that are available in this language at a much larger scale than most other languages in the same family, notably a parallel corpus with English. Since its formation in 1999, the Legislative Assembly of Nunavut has been publishing its proceedings (known as a Hansard) in both Inuktitut (`iku`) and English.[3] In the subsequent 20 years, the collected Nunavut Hansard has grown to be a substantial bilingual corpus (Martin et al., 2003, 2005; Farley, 2008; Joanis et al., 2020), putting Inuktitut in the perhaps unique position of a polysynthetic language with a parallel corpus of more than a million sentence pairs. We discuss the different versions of this data, and their preprocessing for machine translation, in Section 4.2.

  We also made use of a Inuktitut translation[4] of the Bible for language modelling experiments. We decided to exclude the Hansard in the language modelling experiments as including it would make the Inuktitut dataset

---

[1]`https://live.bible.is/bible/ESSWYI`
[2]`bibles.org`
[3]It should be noted that Legislative Assembly of Nunavut discourse takes place in several Inuktut varieties, as well as English; a more detailed description of the construction and dialect situation of the Hansard will be available in Joanis et al. (2020).
[4]`bible.com`

Language	Code	Corpus	Sentences	Tokens	Types	TTR	MDN
Central Alaskan Yup'ik	esu	Bible	59575	566544	138320	0.244	3.86
English	eng	Bible	62049	1057713	22201	0.021	42.90
Chukchi	ckt	Transcripts	1015	5309	2387	0.450	2.22
Inuktitut	iku	Bible	31103	459571	126165	0.275	3.64
Inuktitut	iku	Hansard	1300148	10869995	1563883	0.144	6.95
English	eng	Hansard	1300148	20367595	59234	0.003	343.81
Guaraní	grn	Bible	30078	629099	45766	0.073	12.71
Spanish	spa	Bible	30078	822192	31625	0.038	23.75
St. Lawrence Island Yupik	ess	Books	24456	214090	60414	0.282	3.32
St. Lawrence Island Yupik	ess	New Testament	8002	119482	32532	0.272	3.45
English	eng	New Testament	8002	273064	9071	0.033	28.37

Table 3.2: Statistics of the written corpora, including type-token ratio (TTR) and mean distance to next novel type (MDN).

substantially different from other datasets and thus making it hard to compare it with other languages. How we preprocessed the data for language modelling is discussed in Chapter 5.

3.1.5   Crow

Crow (Apsáalooke, language code cro) is one of the most widely spoken languages of the Siouan family, with approximately 3500 speakers. The Siouan languages are native primarily to the Great Plains of North America, and Crow specifically is spoken in southern Montana.

Our primary resource for Crow was a series of audio recordings for a dictionary developed by the Language Conservancy, an organization that protects and revitalizes Native American languages. This corpus consists of 11.7 hours of recordings produced by 14 speakers. The data is entirely composed of single words and short phrases from the online Crow Dictionary project (The Crow Language Conservancy, 2019). This data was obtained on special permission from the Language Conservancy and is not publicly available.

3.1.6   Guaraní

Guaraní (grn) is a Tupian language native to South America. It is an official language of Paraguay and the most widely spoken language in the country with almost 5 million speakers. It is also the only indigenous language of the Americas with a large number of non-indigenous native speakers.

We were able to obtain Guaraní-Spanish parallel Bible translations. The Guaraní Bible was translated and published by the Sociedad Bíblica Paraguaya. The parallel translations were used for language modelling and machine translation experiments. A morphological analyser developed by Kuznetsova and Tyers (2019), apertium-grn, was also used.

3.2   Descriptive statistics of the corpora

The polysynthetic languages described above differ significantly from languages such as English and Spanish. One major point of difference is in the ratio of word types to word tokens; given the number of word tokens and the number of unique word types, the type-token ratio is calculated as $\text{TTR} = \frac{|types|}{|tokens|}$. Another useful metric, proposed by Hasegawa-Johnson et al. (2017a) and used for polysynthetic language by Schwartz et al. (2020), calculates the mean distance to the next novel word type (MDN).

Table 3.2 displays these text metrics for all textual corpora used. Large differences exist between different languages and between different corpora of the same language with respect to these metrics. The polysynthetic languages examined display higher type-token ratios and lower average distances to the next novel word type in comparison to the non-polysynthetic languages (English and Spanish). This is particularly poignant for parallel corpora. The New Testament in English has a type-token ratio approximately nine times lower than St. Lawrence Island Yupik. This is somewhat expected as the central focus of this work is determining effective strategies for

---

Algorithm 1: Mean distance to next novel type

Result: Mean distance to next novel type
distances = list;
types = list;
current_distance = 0; **for** *word in text* **do**
    if *word in types* **then**
       | current_distance++;
    end
    else
       | distances.append(current_distance);
       | current_distance = 0;
    end
end
distance = avg(distances)

---

working with highly morphologically complex polysynthetic languages and previous research (Kettunen, 2014) has indicated that morphological complexity is correlated with metrics like TTR.

The datasets utilized cover a large number of different domains as well, including religious texts, parlimentary proceedings, audio transcriptions, and data scraped from internet resources. These domain differences contribute to the differences in corpus properties as well. For example, both the English Bible and the English Nunavut Hansard corpus have lower type token ratios and higher mean distances to the next novel type. However, the formulaic language of parliamentary proceedings causes the English Hansard corpus to have a type-token ratio seven times lower than the English Bible used. These domain differences were controlled for the language modelling experiments described in Chapter 5 by using the New Testament for several different languages. For the other tasks, comparisons between languages are used sparingly if similar genres of text are not available for both languages.

3.3   Preprocessing

We preprocessed the corpora for 1) machine translation and 2) language modelling experiments. The general principle and strategies we adapted for preprocessing for both experiments are very similar. We removed any redundant lines and verse numbers to clean up the corpora. We made sure to normalize apostrophes so that they remained as part of a word after we tokenized the data using Moses scripts (Koehn et al., 2007). As truecasing is a common practice in machine translation, we truecased the text for machine translation experiments, but not for language modelling experiments. Using the cleaned-up datasets, we explored different tokenization strategies. FST and BPE segmentation methods were adapted for machine translation experiments, and character, BPE, Morfessor and FST segmentation levels were used for language modelling experiments. Details about how we selected and preprocessed the datasets for the two sets of experiments are discussed in Chapter 4 (Machine Translation) and Chapter 5 (Language modelling), respectively.

3.4   Estimating weights for finite-state morphological analyzers

We used three approaches to estimate weights for our finite-state analysers, one supervised, one heuristic and one unsupervised. The supervised method was the most simple. We had a small corpus of annotated (manually disambiguated) text for Guaraní, the test corpus from Kuznetsova and Tyers (2019). We used this and assigned a weight to all wordform:analyses pairs of 1. For the wordform-analysis pairs found in the corpus, a weight was assigned equal to $1 - P(a|w)$, where $P(a|w)$ is the number of times the analysis occurs with the particular wordform over the total number of times the wordform appears. This is necessarily a number between zero and one and thus for wordforms seen in the corpus, their analysis received a lower weight than unseen wordform-analysis pairs. Given the size of the corpus, 2020 wordforms, the majority of the wordforms seen in the corpora were unseen. For both the Yupik analyser and the Guaraní analyser we added an additional heuristic, for each

morpheme boundary, we increased the weight by 1. The motivation behind this heuristic is that we wanted to favor lexicalized forms and defavor forms with very many derivations when there was a lexicalized alternative. In addition, we experimented with a novel unsupervised approach to weighting the transducers based on byte-pair encoding (BPE; Sennrich et al., 2016).

Chapter 4

Machine Translation

4.1 Introduction

This chapter discusses neural machine translation (NMT) experiments for translation into, out of, and between polysynthetic languages. While polysynthetic and, more generally, morphologically complex languages are often considered to pose a greater challenge for machine translation research than languages with relatively simple morphology (Oflazer and Durgar El-Kahlout, 2007; Bojar et al., 2015), this challenge is often entangled with the challenges of low-resource machine translation. What really causes this challenge? Is it the length and complexity of the word forms? The type-token ratio and data sparsity? A lack of sufficient training data or a need for more training data than morphologically simple languages? A matter of many evaluation metrics being ill-suited to morphologically complex languages? Some combination of all of this?

In this work, we take steps towards answering two relevant questions through experiments on machine translation between English, Inuktitut, and Yupik as well as Guaraní and Spanish. First, can we untangle the influences of small data and morphological complexity on the challenge of modelling these languages? Second, can we make use of higher-resource languages in the same language family to improve machine translation of lower-resource languages? We examine the first through translation of Inuktitut using a new, larger, pre-release version of the Nunavut Hansard,[1] as described in Sections 3.1.4, 4.2.1 and 4.3. We examine the second through a series of experiments on low-resource machine translation (described in Section 4.4); our most promising experiments incorporate Inuktitut data into the translation of Yupik data (Table 4.8).

We first discuss the data resources for machine translation, providing more detail about data size, preprocessing, and the like (Section 4.2). This is followed by descriptions of our machine translation experiments. Section 4.3.3 briefly covers challenges of machine translation evaluation for polysynthetic languages.

The main contributions of our machine translation work during this workshop are as follows.

- We achieved state-of-the-art performance on translation between Inuktitut and English (since surpassed by Joanis et al. (2020)).

- With first access to the beta version 3.0 of the Nunavut Hansard (Joanis et al., 2020), we were able to provide feedback and best practices for preprocessing the dataset and contributed to knowledge about existing character and spelling variations in the dataset.

- We collected empirical evidence on several well-known but unresolved challenges, such as best practices in token segmentation for MT into and out of polysynthetic languages, as well as an examination of how to evaluate MT into polysynthetic languages.

- We successfully used multilingual neural machine translation methods to improve translation quality into low-resource languages using data from related languages. Notably, our "low-resource" languages were lower resource than much of the literature, and we produced improvements without the use of large monolingual corpora (which are unavailable for these languages and many other languages of interest). We observed these improvements across both n-gram-oriented and semantic-oriented metrics.

---

[1] While this was a pre-release at the time of this workshop, the data has now been made available publicly; see Joanis et al. (2020).

4.2   Parallel Data Resources

Chapter 3 describes the general data resources used throughout the workshop. Here we provide a more in-depth look at the resources used for machine translation specifically, including some notes on preprocessing.

	Train	Dev.	Test
`iku-eng`	1300148	3088	2780
`ess-eng`	5838	1142	1750
`esu-eng`	30724	1279	927
`grn-spa`	28050	1129	875

Table 4.1: Preprocessed lines of parallel training, development/validation, and test data for machine translation experiments.

The machine translation resources available to us ranged from moderate to extremely low resource, as shown in Table 4.1.

4.2.1   Inuktitut–English Data

As described is Section 3.1.4, there have been several releases of the Nunavut Hansard. The first, version 1.0, was released to the natural language processing community in Martin et al. (2003), and consisted of 3.4 million English tokens and 1.6 million Inuktitut tokens of parallel data. A subsequent update, version 1.1, corrected some errors in version 1.0 (Martin et al., 2005). Version 2.0 covered proceedings from 1999 through late 2007 (excluding 2003) with about 5.5 million English tokens and 2.6 million Inuktitut tokens (Farley, 2008).

For the purposes of this workshop, we received pre-release access to a beta version of the Nunavut Hansard Inuktitut–English parallel corpus version 3.0, which contains 17.3 million English tokens and 8.1 million Inuktitut tokens, a huge increase over the original data. We refer to this pre-release version as 3.0 or 3.0 beta. We use deduplicated development and test sets in our experiments. The final Nunavut Hansard Inuktitut–English parallel corpus version 3.0 corpus is now available and is described in Joanis et al. (2020). Through our early access to this corpus, we provided feedback on the corpus and on preprocessing best practices, which have been incorporated into the data release.

The corpus contains 17.3 million English tokens and 8.1 million Inuktitut tokens, spanning 1999 to 2017, a major increase over the version 1.0 and 2.0 releases (Martin et al., 2003, 2005; Farley, 2008). This is the largest corpus we had access to for this workshop, and is arguably no longer truly "low-resource" for machine translation research. It is, however very domain-specific, and differs in domain from the other parallel corpora we use in our experiments.

As prior machine translation work performed translation on romanized Inuktitut (Micher, 2018b), we chose to do the same. We converted Inuktitut data from syllabics as follows: we first applied `uniconv`,[2] then repaired errors (e.g., incorrectly handled accented French characters in the Inuktitut data) using `iconv`, then identified and corrected other characters using a hand-built preprocessing script (including treating word-internal apostrophes as non-breaking characters on the Inuktitut side of the data).[3]

We ran standard preprocessing scripts from Moses (Koehn et al., 2007): punctuation normalization, tokenization, cleaning, and truecasing. We discuss subword segmentation in Section 4.3.

4.2.2   Yupik–English Data

We had access to parallel data for two Yupik languages: St. Lawrence Island Yupik (`ess`) and Central Alaskan Yup'ik (`esu`). In both cases, all of the available data was verse-aligned data drawn from the Bible. For St. Lawrence Island Yupik, we had access to New Testament data only. We used Luke for development and validation and used John for testing. The remainder of the data was used for training. The data was preprocessed for machine translation experiments as follows: bracketed text was removed from the English data,[4] apostrophes were normalized in

---

[2]`uniconv` is distributed with Yudit: `www.yudit.org`

[3]Joanis et al. (2020) provides slightly updated scripts; we note that neither those scripts nor the ones described here fully conform to spelling and romanization conventions as described in the Nunavut Utilities (`www.gov.nu.ca/culture-and-heritage/information/computer-tools`).

[4]This consisted of rephrasings of entire verses, and was not present in all verses.

St. Lawrence Island Yupik, and then all data was punctuation-normalized, tokenized, cleaned, and truecased using standard Moses scripts (Koehn et al., 2007) with English default settings.

For Central Alaskan Yup'ik, we had access to the full Bible. For consistency, we still used Luke for development and validation and used John for testing. The remainder of the data was used for training. For Central Alaskan Yup'ik, we normalize apostrophes and convert characters with certain diacritics that would otherwise be split by the Moses tokenizer. Both Central Alaskan Yup'ik and its corresponding English translations were punctuation-normalized, tokenized, cleaned, and truecased using standard Moses scripts (Koehn et al., 2007) with English default settings. In the case of Central Alaskan Yup'ik, we performed tokenization without aggressive hyphen-splitting.[5]

Table 4.1 shows the number of lines in the datasets; the Central Alaskan Yup'ik training data is more than 5 times larger than the St. Lawrence Island Yupik training data.

4.2.3 Guaraní–Spanish Data

As with the Yupik datasets, we had verse-aligned parallel Bible data available in Spanish and Guaraní. We used Luke for development and validation and used John for testing, with the remaining data used for training. Guaraní data was first preprocessed with quotation and apostrophe normalization, along with the removal of paragraph and other symbols that were artifacts of the initial data creation. Guaraní and Spanish data were then punctuation-normalized, tokenized, cleaned, and truecased using standard Moses scripts (Koehn et al., 2007) using Spanish defaults.

4.3 Inuktitut Machine Translation Experiments

Our Inuktitut-English machine translation efforts were largely concerned with doing initial experiments on the pre-release version of the Nunavut Hansard parallel corpus. Being substantially larger than previous releases – to our knowledge, by far the largest aligned parallel corpus of a polysynthetic language to date – this corpus offered a unique opportunity to try contemporary NMT methods on Inuktitut, and consider whether methods of segmentation like byte-pair encoding (BPE; Sennrich et al., 2016) are sufficient to handle a language of this level of complexity.

In the experiments that follow, our baseline systems – that is, conventional Transformer (Vaswani et al., 2017) NMT systems, using BPE and standard hyperparameter settings – always outperformed the experimental systems (which included special segmentation procedures and multi-source attention). This suggests that contemporary methods are indeed adequate for processing Inuktitut, although we do not consider the case closed as there are many interesting possibilities for principled segmentation that we have not yet explored.

4.3.1 Segmentation experiments

In this set of experiments, we contrast automatic segmentation (by byte-pair encoding) with more morphological segmentations based on human knowledge of Inuktitut morphology, and also consider a simple method of combining them. We perform our machine translation experiments contrasting these approaches in the Inuktitut-to-English direction.

Byte-Pair Encoding

Byte-pair encoding (BPE; Sennrich et al., 2016) – broadly, the segmentation of text at the character-level into larger chunks by compressing the text and using the resulting compression units as word segmentation – has become a ubiquitous practice in current machine translation. While the units discovered are not guaranteed to correspond to *morphemes* as such, the resulting systems do end up working at a more morpheme-like level, with units larger than a character but smaller than a word.

Table 4.2 shows the segmentation of several words according to four BPE vocabulary sizes. The Inuktitut loanword *siipiisiikkut* (meaning *CBC* or *Canadian Broadcasting Corporation*) is frequent enough in the corpus

---

[5]This keeps hyphenated suffixes attached, but has the downside of non-ideal interactions with subword segmentation, occasionally breaking suffixed biblical names into two parts, with the latter attached to the hyphen and Central Alaskan Yup'ik suffix.

that at 30000 merges it is represented as a single token. The word *qimirruvita* (meaning *are we looking at*, as in the context *Are we looking at trying to find out?* or *qimirruvita qaujimanittinnuk*) can be split into the morpheme *qimirru-* (*to scan, to inspect*[6]) and the verb ending *-vita?* (*are we (3+) ...?*[7]); we see that here BPE successfully respects the morpheme boundary at all sizes, segmenting exactly and only along that boundary with a vocabulary of 30000. For *utaqqivita* (meaning *are we waiting for?*, as in the context *What are we waiting for?* or *kisumik utaqqivita?*), the story is somewhat different. Though the word contains the same suffix *-vita?* with the verb root *utaqqi-* (*to wait*[8]), BPE does not segment the words along the expected morpheme boundaries; the only segmentation that respects them (500) appears to oversegment. In these examples, we are able to see clear morpheme splits in the surface form, but this is not always the case. In many cases, the underlying forms may undergo phonological changes at the boundaries where two morphemes meet, making it impossible to segment the word such that the resulting units have a uniform representation across all examples of that morpheme.

BPE vocab	siipiisiikkut	qimirruvita	utaqqivita
500	si | i | pi | i | si | i | kkut	qi | mi | r | ru | vi | ta	uta | qq | i | vi | ta
5000	si | i | pii | si | i | kkut	qimirru | vi | ta	utaqq | ivi | ta
15000	siipii | si | ikkut	qimirru | vi | ta	utaqqivi | ta
30000	siipiisiikkut	qimirru | vita	utaqqivi | ta

Table 4.2: Segmentation of three words according to BPE at four different vocabulary sizes.

One of our topics of investigation was whether this procedure alone would be sufficient to pre-process Inuktitut for machine translation, whether more sophisticated morphological processing would be necessary, or whether a combination of the two (morphological processing where possible, BPE for the rest) might prevail.

Morphological Analysis

The Nunavut Hansard version 1.1 was the starting point for morphological analysis of the larger, later-released corpus (version 3.0). As version 1.1 is a subset of the days of debate included in version 3.0, we made use of prior morphological processing of version 1.1 when possible (processing described in Micher (2018a) and summarized here). Every word type of the version 1.1 corpus was processed with the Uqailaut analyzer (Farley, 2009), which provides morpheme segmentation and labeling (including deep representation and morphological category tags). About 70% of the corpus was analyzable by this tool. The remaining 30% was subsequently processed using a neural morphological analyzer, which is trained on a subset of the Uqailaut processed data (Micher, 2017). Filtering out noise (concatenations of numbers and alphanumerics), we were left with approximately 413K processed word types from version 1.1 of corpus.

We then extracted the word types from the larger corpus, using the same noise filtering script as with version 1.1 and omitting the word types that had been successfully processed already from version 1.1. We ended up with ~1.14M additional types. From these another ~9K words were identified as English and removed, yielding ~1.13M types to process. However, we note a few differences between these corpora, which affected the processing pipeline. First, the romanization scheme performed for version 1.1 of the Hansard is not identical to the romanization we performed on version 3.0 beta. In many cases, the resulting romanizations of words match, but in the cases that do not, the morphological analysis needed to be performed anew. For example, there are differences in romanization between Hansard versions (e.g. "lh" vs. "&" for the lateral fricative) and between dialects (e.g. "s" vs. "h" for a particular phoneme); since Uqailaut presumes "&" and "h", these are substituted before re-processing. After all of the pre-processing, we followed the same procedure as with version 1.1 of the corpus, first processing what the Uqailaut analyzer would process, and sending the remaining types through the neural morphological analyzer. In total, we have 1,548,500 types, processed through one or the other analyzer.

For our work during the workshop, however, we are training and evaluating using only the Uqailaut segmentations (that is to say, without using the neural parser), as the neural parses were not yet finished at the time of these experiments. We expect that the more complete analyses of the neural parser will have a more positive effect on downstream performance in future experiments.

---

[6]https://uqausiit.ca/node/10333
[7]https://uqausiit.ca/verb-ending/vita
[8]https://uqausiit.ca/node/12189

In the following experiments, the morphologically processed text uses "deep" forms, in the sense of Micher (2017), rather than the surface forms. Since Uqailaut, and thus the neural generalization of it, only parse surface words into deep forms (and do not generate surface words from deep forms), we present our experiments with different segmentation approaches solely in the Inuktitut to English translation direction.

System configuration

The model uses a 3-layer encoder, a 3-layer decoder, a model dimension of 512 and 2048 hidden units in the feed-forward networks. The network was optimized using Adam (Kingma and Ba, 2014), with an initial learning rate of $1\mathrm{e}{-}4$, decreasing by a factor of 0.7 each time the development set BLEU did not improve for 8000 updates, and stopping early when BLEU did not improve for $32,000$ updates.

In addition to the most common automatic MT evaluation metric, BLEU[9] (Papineni et al., 2002), we also evaluated our MT experiments using two recently proposed metrics, chrF[10] (Popović, 2015) and YiSi (Lo, 2019), which were shown to correlate better with human judgments on translation quality in English by Ma et al. (2019).

Results

`iku` segmentation	`eng` segmentation	BLEU	chrF	YiSi-0	YiSi-1
5000 BPE	5000 BPE	**27.7**	**47.1**	**62.9**	**70.8**
Morphological	5000 BPE	23.3	42.5	58.2	66.1
Morph + 5000 BPE	5000 BPE	26.6	46.8	62.6	70.5

Table 4.3: Results of Inuktitut-to-English NMT systems as evaluated by BLEU, chrF, YiSi-0 and YiSi-1 .

We compared BPE of various vocabulary sizes to the morphological analysis described above. In Table 4.3, we observe that morphological analysis underperforms BPE across all metrics.

We think this is not due to a problem in the morphological analysis itself (e.g. identifying morphemes incorrectly), but that the process left unanalyzable words intact, whereas BPE manages to segment all words into more manageable pieces. We therefore also performed a preliminary attempt to combine them, in hopes of combining some of the benefits of true morphological analysis with the statistical advantages of BPE. First, we took the output of morphological analysis (i.e., the input corpus to the "Morphological" system in Table 4.3), trained a new BPE model on it, and segmented it according to this model. Manual inspection of the results of this process suggest that morphemes identified in morphological analysis were typically left intact by BPE – that is to say, they were identified as units by BPE as well – and only unanalyzed words were further segmented.

This system also underperformed the BPE-only system, but only by small margins. We think that this avenue is still promising, as there are many possible ways to integrate BPE and morphology. Many questions remain:

- Does one resegment only the unanalyzed words, or all words?

- Does one *train* the BPE model on only unanalyzed words, or all words?

- Do we use surface morphemes or underlying morphemes?

- Do we rejoin underlying forms or keep them segmented?[11]

Also, as not all the corpus was fully analyzed, more development in neural analysis will probably lead to improvements downstream.

---

[9]BLEU scores were computed using SacreBLEU (Post, 2018), compared to untokenized but punctuation-normalized references. `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.2`

[10]chrF scores were computed against untokenized but punctuation-normalized references using SacreBLEU with `chrF2+case.mixed+numchars.6+numrefs.1+space.False+version.1.4.2` settings.

[11]Joanis et al. (2020) finds that using underlying forms, but rejoining them before BPE segmentation, gives a performance improvement over deep forms alone in corpus alignment.

4.3.2   Single source and multi-source experiments

One experimental theme we pursued in this workshop was whether multi-source techniques (Zoph and Knight, 2016; Nishimura et al., 2018; Libovický and Helcl, 2017), typically used for training MT systems with multiple source languages, could be of value when applied to multiple *representations* of the input text, as a potential way to combine the benefits of two different kinds of analysis.

A recent result in multilingual machine translation (Littell et al., 2019) suggested that it can be valuable, when training MT on a corpus that has undergone significant processing (in that case, machine translation of the original source into Russian), to attend to *both* the original text and its processed version. That is to say, "attention" in MT makes it possible to avoid having to choose between using the original text or a process that may have been helpful (or may have destroyed useful information); rather, we can allow the model to attend to the results of any stage in the pipeline, and learn for itself which representations to attend to the most. The above result concerned a pre-processing step that was itself machine translation – that is to say, this was a "pivot" system in which L1 is translated to L2, and L2 is translated into L3. We were wondering whether the result might also apply for processing steps that were *not* machine translation. Would, for example, it be fruitful to attend to two different pre-processings: say, BPE and morphological, syllabics or romanized, etc.?

System configuration

The following experiments were performed using the architecture in Littell et al. (2019), a variant of Transformer (Vaswani et al., 2017) with multi-source attention, implemented in the Sockeye framework (Hieber et al., 2017) for machine translation.

The model uses two 3-layer encoders (one for each source type), a 3-layer decoder, a model dimension of 512 and 2048 hidden units in the feed-forward networks. The decoder attended to each decoder using "flat" attention (that is, attending to each and combining the result by simple addition, rather than an additional, hierarchical attention layer). The network was optimized using Adam (Kingma and Ba, 2014), with an initial learning rate of $1e-4$, decreasing by a factor of 0.7 each time the development set BLEU did not improve for 8000 updates, and stopping early when BLEU did not improve for $32,000$ updates.

Results

As an initial sanity check, we performed two tests of the idea:

1. Source 1: BPE vocab size 5000, source 2: BPE vocab size 30000

2. Source 1: Inuktitut in syllabics, BPE vocab size 5000; source 2: Inuktitut romanized, BPE vocab size 5000.

We did not expect these to show significant gains, but we wanted to make sure the systems did not experience a serious drop in scores. Unfortunately, Table 4.4 indeed showed such a performance drop, with the multi-source systems performing very poorly.

Source	Target	BLEU
Inuktitut, syllabics, BPE 5000	English, BPE 5000	30.3
Inuktitut, romanized, BPE 5000	English, BPE 5000	27.7
Inuktitut, syllabics, BPE 5000 + Inuktitut, romanized, BPE 5000	English, BPE 5000	6.3
Inuktitut, romanized, BPE 5000 + Inuktitut, romanized, BPE 30000	English, BPE 5000	2.5

Table 4.4: Preliminary multi-source `iku`→`eng`

We believe this is because the multi-source source system greatly increases the number of parameters without an associated increase in information in the corpus. If we compare this to the positive results in Littell et al. (2019), the difference is that there the introduction of a third language greatly increases the amount of information available to the system: it is not just another view of the same data. So, rather than continue exploring additional monolingual multi-source setups (e.g., BPE and morphology together), we instead moved on to the multilingual multi-source experiments detailed in Section 4.4.2.

4.3.3 Challenges in Evaluation of English-to-Inuktitut MT

For questions of segmentation, we primarily looked at the Inuktitut-to-English direction, since our morphological analyzer was only able to parse, rather than generate. (That is to say, while we could output segmented, underlying morphemes, we could not, at that time, rejoin them into fluent outputs.) For English-to-Inuktitut, we only looked at BPE-based systems, since these can trivially be de-segmented. In this translation direction, we focused on questions of evaluation because morphologically complex languages pose a challenge in terms of the choice of automatic evaluation metric.

BLEU (Papineni et al., 2002) is a common metric for evaluation of machine translation output given reference translations. However, because BLEU score is (typically) computed at the word level, an error in a single morpheme is penalized just as harshly as a completely incorrect choice of terminology. This can be expected to have a particularly detrimental effect when evaluating translation output in morphologically complex languages; even if the system chooses the correct lemma, any errors of morphological inflection will be counted as whole-word errors, decreasing the count of correctly-predicted n-grams. BLEU score could also be computed over byte pair encodings rather than words, but this poses challenges when trying to compare systems built with different vocabularies.

chrF sidesteps the segmentation issue by first removing whitespace before counting character n-grams and computes a precision/recall-balanced score over the character n-gram counts. On the other hand, YiSi-0 respects the word boundaries in the MT output but uses the character-level longest common substring accuracy to evaluate the word-level similarities and aggregates the word-level similarity scores into the sentence-level score. These two automatic evaluation metrics based on character-level information would be more suitable for evaluating MT output in morphological complex languages. In fact, Ma et al. (2018) showed that chrF correlates the best with human in evaluating Finnish translation output and YiSi-0 correlates the best with human in evaluating Turkish translation output. However, we think it important to point out that the complexity of Inuktitut morphology is higher than that of Finnish or Turkish and there is no existing work on MT evaluation for polysynthetic languages. This remains an area for future work.

System configuration

The English-to-Inuktitut MT system was built using the same architecture as that of the system mentioned in Section 4.3.1. We evaluated the system at both word-level and 5k BPE-vocabulary segmentation using BLEU,[12] chrF,[13] and YiSi-0. Since YiSi-0 is a weighted harmonic mean of precision and recall, we also dissected YiSi-0 into pure precision and recall for further analysis.

Results

First and the foremost, we have to emphasize that MT system scores for different translation directions are not directly comparable. Thus, one should *not* conclude from Table 4.5 that translating Inuktitut into English is an easier task to the opposite direction, or the translation quality of a system in one direction is better than that in the other direction.

Translation direction	Evaluation unit	BLEU	chrF	YiSi-0		
				weighted-F	precision	recall
iku→eng	word	27.7	47.1	62.9	66.2	62.1
eng→iku	word	17.8	46.7	48.0	49.9	47.9
iku→eng	5000 BPE	29.5	47.4	64.1	67.6	63.3
eng→iku	5000 BPE	13.7	46.4	56.4	59.0	56.0

Table 4.5: Results of English-to-Inuktitut NMT systems as evaluated by BLEU, chrF and YiSi-0 (with pure YiSi-0 precision, i.e. α=0.0 and recall, i.e. α=1.0 for supplementary analysis).

---

[12]In this table and table 4.4, BLEU scores were computed against untokenized but punctuation-normalized references using SacreBLEU with `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.2` settings.

[13]chrF scores were computed against untokenized but punctuation-normalized references using SacreBLEU with `chrF2+case.mixed+numchars.6+numrefs.1+space.False+version.1.4.2` settings.

Instead, we would like to point out that there is a notable difference in word-level BLEU scores for the systems in two translation directions because BLEU penalizes systems on failing to correctly inflect a word form equally harshly as choosing an entirely incorrect word; thus MT systems translating into morphological complex languages are expected to achieve lower word-level BLEU scores. A huge difference can also be seen in YiSi-0 scores using word segmentation in evaluation. However, the chrF score difference between the two translation directions is marginal.

When evaluating translation output at subword unit level, both BLEU and chrF showed a wider score difference when the translation direction was flipped. However, YiSi-0 showed a smaller difference. The contradicting results showed that evaluating translation output in polysynthetic languages itself is a challenging and unsolved research problem.

Without human evaluation on translation output in polysynthetic languages, we could not conclude whether the quality of the English-to-Inuktitut MT system is acceptable or not (or whether it is sufficient for some use cases but not others). We hope that future human evaluation of machine translation into polysynthetic languages will provide a basis for the examination of different evaluation approaches, allowing future researchers to select the most appropriate evaluation metrics.

4.4 Low-Resource Experiments

In keeping with the theme of the workshop, our low-resource machine translation experiments involve neural systems rather than phrase-based ones, despite the fact that they are built from extremely small datasets. While we perform our experiments with fairly simple modern neural models and minimal hyperparameter tuning, recent work (Sennrich and Zhang, 2019) suggests that careful tuning of hyperparameters can result in NMT systems outperforming statistical machine translation systems even on datasets of around 5000 sentences (comparable to our smaller datasets).

Most of the low-resource machine translation experiments were performed using Sockeye (Hieber et al., 2017), and the multi-source generalization of Sockeye introduced in Littell et al. (2019).

4.4.1 Baselines and Vocabularies

	RNN BPE	Transf. BPE	Transf. Word	Transf. FST	Transf. FST+BPE
ess→eng	4.2	**8.4**	7.3		
eng→ess	3.3	**4.4**	3.5		
esu→eng	10.7	**13.9**	6.5		
eng→esu	**5.4**	5.3	3.3		
grn→spa		**10.5**	7.4	7.1	9.6
spa→grn		**8.6**	7.1	8.3	8.1

Table 4.6: BLEU scores of baseline and vocabulary experiments for Yupik–English and Guaraní–Spanish machine translation experiments. All BPE vocabularies in this table are of size 5000, learned separately.

We first compare RNN and Transformer translation models using BPE vocabularies of 5000. The size of 5000 was selected for consistency with other experiments and because it was among the highest performing vocabulary size on initial RNN experiments for several language pairs (not reported here). The RNN models were trained using OpenNMT (Klein et al., 2017) with default settings, and the Transformer models were trained using Sockeye (Hieber et al., 2017) with a 3 layer encoder, 3 layer decoder, batch size 2048, optimized toward perplexity, and the remaining parameters set to defaults. As Table 4.6 shows, the Transformer system outperformed the RNN system in all but one case (which was within 0.1 BLEU); we use the Transformer system for all remaining experiments.

We compare using a BPE vocabulary of 5000 symbols to using a whole word vocabulary. In all cases, the BPE vocabulary outperforms the whole word vocabulary (by between 0.9 and 7.4 BLEU points). Using whole words, English–St. Lawrence Island Yupik experiments were run with vocabulary sizes of 4787 and 26888 (respectively, including special characters), while English–Central Alaskan Yup'ik whole word vocabularies consisted of 13501

and 106736 types respectively. Given the small data sizes and large Yupik vocabulary sizes, it is unsurprising that BPE outperforms whole words; there may simply not be enough examples of many types in the long tail for the system to accurately translate them, and the word system includes a large number of out of vocabulary items in the test set.

Following the results of the Yupik experiments, we omit the RNN experiments for Guaraní–Spanish and instead start with a baseline of a Transformer model (3 layer encoder, 3 layer decoder, batch size 2048, optimized toward perplexity, remaining parameters set to defaults), using separately learned BPE encodings for Spanish and Guaraní with vocabularies of 5000 types each. There does exist other work on machine translation for Guaraní–Spanish, notably an online gister[14] and work in Rudnick (2018). Though Rudnick (2018) also performs experiments on Bible translation, we do not compare directly, as those results are measured on stemmed output.

For Guaraní–Spanish, we also experiment with full-word vocabularies, FST-segmented vocabulary (Guaraní side only; Spanish side BPE 5000), and an FST-segmented vocabulary with backoff to BPE (all Guaraní words left unsegmented by the FST were segmented by a BPE model learned for a BPE 5000 vocabulary on Guaraní; Spanish side BPE 5000). As shown in Table 4.6, the baseline BPE model outperforms all other experiments.[15]

4.4.2 Yupik Language Experiments

Our Yupik language experiments begin with baseline RNN and Transformer models. Finding that the Transformer strongly outperforms the RNN (Table 4.6), we perform the remainder of the experiments with the Transformer architecture only.

In addition to the baseline, we perform two experiments: multi-source experiments on a multi-parallel subset of the data and multilingual NMT system experiments. BPE vocabularies of size 5000 were learned separately on each language's training data using `subword-nmt` (Sennrich et al., 2016). Our most promising low-resource experiments, described in Section 4.4.2 involve the use of higher resource languages from the same language family to build multilingual neural machine translation systems which can then be finetuned for specific low-resource languages.

Multisource

In order to experiment with multisource machine translation, we build a multiparallel verse-aligned corpus from the intersection of all available Yupik Bible data. The resulting New Testament corpus has 5449 lines for training, 1091 lines for development and validation, and 874 lines for testing. It contains data in St. Lawrence Island Yupik and Central Alaskan Yup'ik, as well as data from two English Bibles. We call the English Bibles eng_{ess} (for the English Bible originally aligned to St. Lawrence Island Yupik) and eng_{esu} (for the English Bible originally aligned to Central Alaskan Yup'ik). We preprocessed them identically to the baseline experiments, with one change: we removed verse numbers from Central Alaskan Yup'ik and its corresponding English (eng_{esu}) as those were not present in the St. Lawrence Island Yupik corpus.

We compared single-source (Sing.) and multi-source (Mult.) approaches, as described in §4.3.2, as well as separately learned and jointly learned 5000 symbol BPE representations (the joint BPE representations were learned across all 4 sides of the multiparallel corpus). For the multi-source experiments, we tried translating into Central Alaskan Yup'ik using its corresponding English and St. Lawrence Island Yupik, as well as translating into St. Lawrence Island Yupik using its corresponding English and Central Alaskan Yup'ik. Without any major parameter search, we found that the joint BPE single-source systems performed the best.

As these BLEU scores are extremely low, it is quite difficult to draw any conclusions from this set of experiments; the following notes should be understood in that context. We do observe that for single-source, using a jointly trained BPE vocabulary performs better than separately trained BPE vocabularies. This may be due in part to improved translation of copied terms (e.g., names). We do not observe the same consistency in multisource. Perhaps unintuitively, in single-source experiments, we find that swapping the English Bibles (translating eng_{esu} into ess and eng_{ess} into esu) performs better than the "correct" pairs. This highlights several challenges of performing machine translation using Bible corpora: we do not have a guarantee in our case that the "source" English Bible is the version from which the Yupik Bibles were translated, Bible translations may rely on metaphor

---

[14] `http://iguarani.com/`
[15] BLEU scores were computed against untokenized but punctuation-normalized references using SacreBLEU with `BLEU+case.lc+numrefs.1+smooth.exp+tok.13.a+version.1.3.7` settings.

	Sing.	Mult.	Sing. (joint)	Mult. (joint)	Mult. (joint+tied)
ess–esu	3.8		4.7		
eng$_{ess}$–esu	4.8		4.9		
eng$_{esu}$–esu	3.6	3.2	3.9	2.8	3.1
eng$_{ess}$–ess	4.0	3.1	4.4	3.4	3.4
eng$_{esu}$–ess	4.7		5.4		
esu–ess	4.2		4.8		

Table 4.7: BLEU score results for experiments on joint and separate BPE learning, along with multisource experiments. Tested on the multiparallel subset of Yupik corpora.

or other non-literal phrases, and verse alignment provides additional challenges due to mismatches between sentence and verse boundaries. In some cases, we observe that a sentence spans more than one verse, with a name appearing in the first verse in English and in the second verse in Yupik or vice versa, an impossible challenge for machine translation without extrasentential context to overcome; this is a known challenge in parallel Bible corpora (Mayer and Cysouw, 2014). We also did not perform hyperparameter optimization due to time constraints; more extensively tuned models may show different results.

Multilingual

Multilingual neural machine translation has been proposed as a means of improving neural machine translation of low-resource languages, using a variety of distinct approaches. These approaches depend are split into approaches to translate into or out of low-resource languages. Neubig and Hu (2018) explore the multilingual translation task translating from multiple low-resource languages into a single high-resource language. Gu et al. (2018) also work in the same translation direction, and incorporate large amounts of monolingual data and many closely-related source languages.

Our interest is on translation *into* low-resource languages. In that direction, Ha et al. (2016) perform multilingual neural machine translation by tagging each subword with a language-specific tag, and then building a system based on available training data. Johnson et al. (2017) use a single special token at the beginning of input sentences to indicate the desired target language to translate into. Rikters et al. (2018) follow this approach to do multilingual translation into and out of morphologically rich languages, though their low-resource setting consists of more than 3 million sentence pairs.

St. Lawrence Island Yupik, Central Alaskan Yup'ik, and Inuktitut belong to the same language family. Despite this, they have very limited vocabulary overlap in our parallel data (less than 1% type overlap between Inuktitut and Yupik, and less than a 3% type overlap between St. Lawrence Island Yupik and Central Alaskan Yup'ik). This is certainly due in part to the different domains we had available: legislative text (Inuktitut) and Bible (Yupik). As described in Section 4.2.2 and Section 4.2.1, our data spans a wide range in terms of size, from approximately 5000 lines of text to approximately 1.3 million lines. We approximately follow the Johnson et al. (2017) approach in our approach to translating from English into Inuktitut and Yupik languages.

	Baseline	Multilingual	ess-Ad. Multi.	esu-Ad. Multi.
eng–ess	4.4	5.8	**6.5**	1.3
eng–esu	5.3	5.7	1.9	**6.0**

Table 4.8: BLEU scores for experiments on multilingual neural machine translation. The baseline is the original Transformer baseline for each language pair. Multilingual is the single multilingual system (trained on Inuktitut and Yupik data), and the remaining two columns show that system fine-tuned on a particular variety of Yupik.

We train joint BPE (vocabulary 5000) on Inuktitut, St. Lawrence Island Yupik, and Central Alaskan Yup'ik, downsampling the Inuktitut and upsampling St. Lawrence Island Yupik to match the size of Central Alaskan Yup'ik. We prepend a language tag (e.g. "<ess>") to each source and target sentence in the three sub-corpora. Next we train a Transformer model (our "multilingual baseline") on the concatenation of all available training

	Baseline	Multilingual	`ess`-Ad. Multi.	`esu`-Ad. Multi.
`eng–ess`	26.9	28.0	**30.1**	10.5
`eng–esu`	31.0	32.5	16.7	**33.2**

Table 4.9: YiSi-1 scores (higher is better) computed using `ess` or `esu` BPE 5000 embeddings built by `word2vec` (Mikolov et al., 2013) for experiments on multilingual neural machine translation. The baseline is the original Transformer baseline for each language pair. Multilingual is the single multilingual system (trained on Inuktitut and Yupik data), and the remaining two columns show that system fine-tuned on a particular variety of Yupik.

data (with no sampling, 3 layer encoder, 3 layer decoder, 512 embedding size, early stopping on perplexity of the concatenated development data). For St. Lawrence Island Yupik and Central Alaskan Yup'ik, we then fine-tune the multilingual baseline on all language-specific training data (with early stopping based on perplexity on the language-specific development data). The BLEU score results are shown in Table 4.8. Table 4.9 reports YiSi results, which follow the same trend as the BLEU score results. As expected, fine-tuning on language specific data boosts performance on that particular language (while the output on the other language appears to exhibit catastrophic forgetting (Kirkpatrick et al., 2017)), giving us our best performance. However, with BLEU scores in the single digits, it is clear that there is still a long way to go before the MT output may be genuinely useful (e.g. in post-editing or interactive translation) for these low-resource languages.

Chapter 5

Language Modelling

In this chapter, we report on language modelling experiments, comparing different tokenization strategies for polysynthetic languages. We trained a state-of-the-art RNN language model using the character, BPE, Morfessor and FST as the unit for segmenting text data. In order to facilitate comparisons across the tokenization strategies, we carefully selected datasets for two experimental settings: 1) A setting where all the data available for a language is used and 2) a setting where only the New Testament in a language is used. The former setting provides us an opportunity to utilize all the data we have in a language while the latter allows us to draw a more precise comparison across languages. We use the average perplexity per character or the character-level perplexity as a metric to compare different models. The results show that the linguistically-oriented, FST segmentation strategy performed the best in modelling polysynthetic languages when it was available. In addition, difficulty of modelling different languages is compared using the average perplexity per word or the word-level perplexity. The potential of FST in aiding language modelling of polysynthetic languages and implications on comparing models for different languages are discussed.

5.1 Data Preparation

After much consideration, we selected four low-resource, polysynthetic languages for our language modelling experiments (hereafter referred to by ISO 639-3 code): St. Lawrence Island Yupik (`ess`), Central Alaskan Yup'ik (`esu`), Inuktitut (`iku`) and Guaraní (`grn`). These languages were chosen because we had the most available text data in them. We had at least the Bible, the Gospel books in New Testament in particular, in these languages, and that allowed us to have a commonality among the datasets to facilitate comparison across the languages. In addition to the polysynthetic languages, we included two well-researched, non-polysynthetic languages: English (`eng`) and Spanish (`spa`). The `eng` and `spa` data were included to provide comparison between polysynthetic languages and non-polysynthetic languages as `esu` and `eng` and `grn` and `spa` were parallel translations.

We designed two experimental settings to fully utilize available data while ensuring comparability across different languages. As for the 1) *all data* setting, we included any available monolingual data in a given language, including but not limited to the New Testament. The second setting, the 2) *New Testament only* setting, focused only on the New Testament data in order to further ensure comparability given the near-parallel data across different languages. Regardless of the settings, Luke was used as the development set and John as the test set to further facilitate fair comparison as we had the Gospel books in all languages. This ensured that different languages

Split	Setting 1: All data	Setting 2: New Testament
Train	Rest of the data available (e.g. Old Testament, transcripts, stories)	Rest of New Testament
Dev	Luke	Luke
Test	John	John

Table 5.1: Train-dev-test split

Language	Sentences	Words	Types	Type/Token	Mean distance to unseen
ess	20,899	206,691	58,637	0.28	3.28
esu	33,102	474,499	106,381	0.22	4.15
iku	31,103	466,705	126,162	0.27	3.70
grn	30,078	622,999	38,944	0.06	14.63
eng	21,835	395,368	11,258	0.03	31.72
spa	30,078	840,937	24,829	0.03	30.39

Table 5.2: Descriptive statistics for setting 1 (all available data)

shared a development set and a test set and a part of the train set (the rest of the New Testament) in common even though the exact train set available in each language may differ from one another. The train set in the 1) *all data* setting included the New Testament, but may also include the Old Testament, transcripts and oral narratives if available. This setting, therefore, fully utilizes the data we had in each language. In the 2) *New Testament* setting, the development and test sets stayed the same, but the train set included the rest of the New Testament only. It should be noted that we did not align the Bibles at the sentence level, and there was some variability among different Bible translations as discussed in Chapter 4. However, esu and eng and grn and spa Bible translations were assumed to be parallel, and we assume that the other Bible translations provide comparable texts with similar intensions overall. While the 2) *New Testament only* setting may provide a more precise comparison, the 1) *all data* setting may be more representative of the reality given the limited size of the data for the former setting. Table 5.1 summarizes the two experimental settings and the dataset split.

Given the data split, we preprocessed the datasets systematically to further ensure comparability among subsequent language models. We removed redundant, bracketed texts when applicable, and normalized apostrophes as they were meaningful in some languages and should not be tokenized separately from their surrounding words. Then, we normalized the punctuation and tokenized the texts using Moses scripts (Koehn et al., 2007) with default settings. The overall preprocessing for language modelling experiments resembles that for machine translation experiments discussed in Chapter 4 except that we did not truecase the data for langauge modelling experiments.

Tables 5.2 and 5.3 summarize descriptive statistics of the preprocessed data under each setting. Overall, it seems that the characteristics of a language as captured by the statistics are quite similar under the two settings. This may not be surprising given that the two settings concern very similar domains. While it remains to be seen if these descriptive statistics would be similar under a different setting for the languages, we observed the followings for the languages given our datasets: As discussed in Chapter 3, the languages seem different in the TTR and mean distance to the next unseen word. ess, esu and iku consistently show a higher TTR and a lower mean distance to the next unseen word than grn. While grn is considered as a polysynthetic language, it seems that grn might be slightly different from the other polysynthetic languages spoken in Alaska (ess, esu, iku). Still, grn is distinctive from spa and eng in that it still had a higher TTR and lower mean distance to the next unseen word. While the spa data seems more complex under the New Testament setting, eng and spa are consistently simpler than polysynthetic languages in terms of TTR and mean distance to the next unseen word.

It is noted that, across languages, the datasets are similar in terms of sentence counts within each experimental setting. While esu-eng and grn-spa differed slightly in terms of the exact sentence count, they are aligned at the verse level. The rest of the data are not aligned at the verse level, but they seem to contain similar number of sentences under the respective data conditions. Note that we did not include the Hansard data for iku. We exclude the data because including it would increase the amount of available data and genre variability for the particular language too much to allow comparison across languages.

Given the similar number of sentences present in each dataset, it is noteworthy that the word count and type count are distinct across the languages. Again, ess, esu and iku seem similar to each other in that they have a smaller number of words and a large number of types than others. This reflects their typological characteristic, that they tend to have longer words with more morphemes, which may lead to more unique tokens. grn still seems distinct from the other polysynthetic languages in that the datasets in the language tend to have more words and less unique words. In fact, grn seems to have similarity with spa in terms of the descriptive statistics even though grn still has a lower mean distance to the next unseen word than spa. eng seems to be clearly more analytic than the other languages as it has more word counts and less type counts.

Language	Sentences	Words	Types	Type/Token	Mean distance to unseen
ess	7,860	121,549	31,928	0.26	3.57
esu	8,464	108,757	30,980	0.28	3.32
iku	7,858	110,977	36,573	0.33	3.03
grn	7,896	171,350	12,779	0.07	12.20
eng	7,870	210,395	5,067	0.02	38.82
spa	7,896	206,707	11,371	0.06	16.01

Table 5.3: Descriptive statistics for Setting 2 (New Testament only)

5.2 Tokenization strategies

We considered five different tokenization strategies in modelling the languages: word, character, BPE, morfessor and FST segmentation methods. In what follows, we briefly explain each tokenization strategy and why they might be helpful in segmenting polysynthetic languages.

5.2.1 Word

A common tokenization strategy is to tokenize text by whitespace or by words. While it may be simple and seem intuitive, this tokenization strategy faces data sparsity and out-of-vocabulary (OOV) issues. For example, if we tokenize by words, *dog* and *dogs* will count as two separate tokens even though there is much shared information between the two. If the train set includes only the singular form and the test set contains only the plural form, the plural form in the test set will be considered as OOV.

(3) aghnaaguq
    aghnagh   -∼:(ng)u   -∼$_f$(g/t)u-   -q
    woman   -to.be   -INTR.IND   -3SG

    'she is a woman' (Jacobson, 2001, p.25-26)

This tokenization method is particularly problematic for polysynthetic languages given their rich morphology. A word in polysynthetic languages may contain several morphemes to express a sentence-like intension. For example, a word in ess, *aghnaaguq*, consists of four morphemes and is translated as *'She is a woman'* as shown in Example (3). Importantly, this results in a high rate of hapax legomena (words appearing only once), which results in much higher OOV rates than observed in most non-polysynthetic languages. In modelling polysynthetic languages, the word-level tokenization is too unrealistic to be useful in predicting the next word, and its performance may be over-estimated or under-estimated depending on how we reward or penalize OOVs. For example, if we do not penalize a model for predicting an OOV symbol for the next word, it may predict an OOV symbol repeatedly for a polysynthetic language to falsely record a good performance. If we do want to penalize OOV, we will have to come up with a metric that does that fairly given our data. Given that the model we adapted did not penalize OOV, we opted to use language models that would not over-generate OOVs.

5.2.2 Character

One possible solution to such issues of word-level tokenization is to tokenize text by the character. The character-level tokenization rarely has OOV issues because a text typically consists of a finite set of characters regardless of its morphological complexity. However, this tokenization method, again, cannot fully utilize the linguistic information present in a text as it reduces all words into a sequence of a finite set of characters. The relationship between *dog* and *dogs* may be easily captured by a character-level model, but words with more complex morphology like Example (3) may be hard to model using the character as the tokenization unit.

While we report our results for character-level models as the baseline to compare other results to, we note that character-level models may not be meaningful for downstream applications for polysynthetic languages such

as keyboard prediction: Predicting a character at a time when a word consists of several morphemes and a long sequence of characters may be too slow or too low-quality.

5.2.3 BPE

If word-level tokenization is too coarse-grained and character-level tokenization is too fine-grained, it may mean that we need to utilize subword units to segment our data. As discussed in Section 4.3.1, byte pair encoding (BPE; Sennrich et al., 2016) is a unsupervised segmentation method that uses subword units. Originally a data compression algorithm (Gage, 1994), BPE has become one of the standard techniques in neural machine translation since Sennrich et al. (2016). Tokens segmented by BPE can represent texts with the minimum entropy by the fixed vocabulary size, which should be chosen as the hyperparameter. BPE segmentation may look like morpheme segmentation for some words, but it is data-driven rather than based on linguistic information. For example, with enough support from a given data, BPE may segment 'lower' as 'low@@ er' (@@ represents a within-word morpheme boundary), which may seem linguistically motivated, but it is also possible to get different segmentations such as 'l@@ ow@@ er' with different hyperparameters and different data conditions. Refer to Table 4.2 for examples of BPE segmentations for machine translation of `iku`, some of which respect morphological boundaries and some of which do not.

We trained a BPE model on the training data and applied the model to all data using `subword-nmt`[1]. We experimented with different vocabulary sizes for BPE segmentation, and report results on two vocabulary sizes: 500 and 5,000. While BPE provides an off-the-shelf method to segment words into subword units, it remains unclear whether the unsupervised method would prove useful in modelling polysynthetic languages.

5.2.4 Morfessor

We adopted another unsupervised segmentation method called Morfessor to compare with BPE. Morfessor is a tool for unsupervised (and semi-supervised) morphological segmentation and has been utilized in speech recognition, MT, and speech retrieval. While there is no literature on its efficiency in neural language modelling tasks for polysynthetic languages, it is said to be useful in modelling languages with rich morphology such as Finnish, Estonian, German and Turkish (Smit et al., 2014). Morfessor uses Maximum a Posteriori (MAP) estimation to approximate morpheme segmentation assuming that a word consists of one or more "morph", yet its results may not be the same as linguistically motivated morpheme segmentation. We used Morfessor 2.0 with the default settings for Morfessor segmentation.

5.2.5 FST segmentation

The last segmentation strategy we considered was segmentation based on FSTs. FST segmentation provides knowledge-based, rule-based segmentation based on linguistic knowledge and analysis. Several FST-based morphological analyzers or morphological segmenters have been developed for polysynthetic languages, and we were able to experiment with two of them for our experiments: `ess` (Chen and Schwartz, 2018) and `grn` (Kuznetsova and Tyers, 2019). The FST-based morphological analyzers produce zero or more morphological analyses for any given word. When there are more than one analysis available for a word, we used heuristics (e.g. choose the shortest analysis) to select one analysis to segment the given word. When there was no analysis available, we used character (character backoff) or BPE (BPE backoff) segmentation for the word. The BPE backoff was performed using the existing BPE segmentations with the vocabulary size of 500 and 5,000. While we were able to obtain this segmentation results only for two polysynthetic languages, this provides a point of comparison between supervised, linguistically motivated segmentation and unsupervised, data-driven segmentation.

5.3 RNN-LSTM

We used a state-of-the-art language model (Merity et al., 2017, 2018) for our language modelling experiments. The RNN model with LSTM has shown to be competitive in modelling English benchmark datasets such as PTB and WikiText-2. We adapted the hyperparameters for WikiText-2 (WT2) with LSTM for Morfessor and FST

---

[1] `https://github.com/rsennrich/subword-nmt`

	Character & BPE	Morfessor & FST
RNN Cell	LSTM	LSTM
Layers	3	3
RNN hidden size	1840	1150
Dropout (e/h/i/o)	0/0.1/0.1/0.4	0.1/0.2/0.65/0.4
Weight drop	0.2	0.5
Weight decay	1.2e-6	1.2e-6
BPTT length	200	70
Batch size	128	80
Input embedding size	400	400
Learning rate	1e-3	30
Epochs	50	200
Random seed	1111	1882
Optimizer	Adam	SGD
LR reduction (lr/10)	[25, 35]	NA

Table 5.4: Hyper-parameters for word- and character-level language modelling experiments

Language	Morfessor	BPE (V=500)	BPE (V=5k)	Character
ess	2.53	2.64	3.34	**2.51**
esu	2.72	2.82	2.84	**2.64**
iku	**2.31**	2.42	2.46	2.36
grn	**2.93**	3.07	3.49	3.03
eng	2.53	2.48	**2.47**	2.51
spa	8.97	2.72	**2.60**	2.69

Table 5.5: Character-level perplexity for setting 1 (all available data). V means the vocabulary size for BPE operation. Bold numbers represent the best model for each language while underlined numbers show the best model for each tokenization.

models and the hyperparameters for character level enwik8 for character and BPE models. Table 5.4 summarizes the hyperparameters.

We acknowledge that none of these models (nor any other models to our knowledge) have been specifically designed to model polysynthetic languages or reported to be used to model polysynthetic languages. With a lack of a language model designed to model polysynthetic languages, we chose a state-of-the-art model that has proven competitive in modelling English instead.

5.4 Character-level perplexity

Perplexity is a measure of language modelling difficulty and calculated by taking the exponent of the average negative log-likelihood per token. Because perplexity as it is depends on the tokenization strategy, we calculate the character-level perplexity for each model to allow comparison among them. We define the character-level perplexity as the exponent of the average negative log-likelihood per character and calculate it by adding up the token-level loss for a given tokenization, multiplying the total loss by the number of tokens in the test set and dividing the value by the number of characters in the test set. We count whitespace and the end of a sentence symbol as separate tokens. This ensures a fair comparison among different tokenization strategies. The choice of character as the common denominator is arbitrary, and it can be other tokenization methods such as the word. Refer to Mielke (2019) for detailed explanations.

Language	Morfessor	BPE (V=500)	BPE (V=5k)	Character
ess	2.77	3.14	3.23	**2.64**
esu	2.98	3.74	3.61	**2.89**
iku	**2.59**	3.02	2.96	2.61
grn	3.16	3.44	3.41	**2.97**
eng	<u>**2.40**</u>	<u>2.81</u>	<u>2.59</u>	<u>2.56</u>
spa	**2.66**	3.23	3.18	2.94

Table 5.6: Character-level perplexity for setting 2 (New Testament only).

5.5   Results & Discussion

Tables 5.5 and 5.6 summarize the language modelling experiment results excluding FST segmentation for the 1) *all data* setting and 2) *New Testament only* setting, respectively. It is suggested that the character and Morfessor models might work better than BPE models for polysynthetic languages. As for the 1) *all data* setting, tokenization by character resulted in the best performance in modelling ess and esu while Morfessor models performed the best for iku and grn. BPE models with the vocabulary size of 5k worked the best with eng and spa. The same trend was observed for the 2) *New Testament* setting for ess, esu and iku: character models performed the best for ess and esu while Morfessor led to the lowest perplexity measure for iku. However, the character-level model resulted in the lowest character-level perplexity for grn while the Morfessor model was the best for eng and spa for the 2) *New Testament* setting. While it is unclear why a certain tokenization method worked better for a language, it is speculated that BPE might not be well-suited in segmenting polysynthetic languages given their morphological richness. A word in a polysynthetic language might consists of several morphemes that are not immediately retrievable based on the surface form. As shown in Example (3), a word in ess may contain a root, a derivational suffix and inflexional suffixes, which may look different in the surface form depending on the morphophonological rules that apply to the suffixation. For example, the derivational suffix (-∼:(ng)u) in example (3) has two morphophonological symbols (∼ and :), the latter of which applies to delete the gh ending of the root (for details see Jacobson, 2001). Given such characteristics of polysynthetic languages, the fact that character models worked the best for ess and esu might mean that those languages were hard to segment with unsupervised segmentation methods like Morfessor and BPE. Segmenting those languages might require getting at the underlying form with linguistically motivated segmentation rather than segmenting the surface form only. Even though Morfessor models worked the best for iku under both settings and for grn under the 1) *all data* setting, the difference between the Morfessor models and character models are quite small.

It should be noted that the hyperparameters for Morfessor and BPE operations are not optimized. While the BPE models with the two hyperparameters (V=500 and V=5k) did not result in the best model for any of the polysynthetic languages, it is possible that different hyperparameters might result in better (or worse) perplexity measures. In a similar note, different datasets in a language might work differently with Morfessor tokenization: the Morfessor segmentation was the best in modelling spa under the 2) *New Testament* only setting, but it was the very worst under the 1) *all data* setting.

Language	Setting	Morfessor	FST				BPE		Character
			-	+BPE(V=500)	+BPE(V=5k)	+char	V=500	V=5k	
ess	All	2.53	**2.30**	2.35	2.36	2.33	2.64	3.34	2.51
ess	NT	2.77	**2.25**	2.41	5.38	2.42	3.14	3.23	2.64
grn	All	2.93	2.74	2.70	2.70	**2.69**	3.07	3.49	3.03
grn	NT	3.16	4.82	2.93	**2.65**	2.68	3.44	3.41	2.97

Table 5.7: Character-level perplexity including FST segmentation

Language	Setting	Morfessor	FST				BPE		Character
			-	+BPE(V=500)	+BPE(V=5k)	+char	V=500	V=5k	
ess	All	1903.37	**882.16**	1037.10	1097.51	980.49	2718.52	18157.32	1790.03
ess	NT	3986.77	**739.92**	1289.70	891605.81	1353.29	10993.00	13975.35	2689.33
grn	All	287.71	203.99	187.76	189.23	**185.96**	372.84	725.33	348.70
grn	NT	432.68	3988.57	288.63	**171.56**	181.21	673.22	644.65	313.79

Table 5.8: Word-level perplexity including FST segmentation

Language	Morfessor	BPE (V=500)	BPE (V=5k)	Character
ess	1903.37	2718.52	18157.32	**1790.03**
esu	2244.44	2969.89	3113.87	**1783.40**
iku	**1469.02**	2185.62	2503.41	1773.08
grn	**287.71**	372.84	725.33	348.70
eng	<u>49.37</u>	<u>45.87</u>	**44.86**	<u>47.83</u>
spa	13051.98	75.10	**62.01**	71.97

Table 5.9: Word-level perplexity for setting 1 (all available data). V denotes vocabulary size for BPE operation

Language	Morfessor	BPE (V=500)	BPE (V=5k)	Character
ess	3986.77	10993.00	13975.35	**2689.34**
esu	4562.23	26323.96	20053.28	**3572.00**
iku	**3923.01**	14970.43	12581.29	4231.44
grn	432.68	673.22	644.65	**313.79**
eng	**39.62**	<u>77.30</u>	<u>55.01</u>	<u>52.03</u>
spa	**67.93**	158.86	148.47	106.16

Table 5.10: Word-level perplexity for setting 2 (New Testament only). V denotes vocabulary size for BPE operation

As a way to utilize rich morphology in modelling polysynthetic languages, we trained FST-based models for `ess` and `grn`. Table 5.7 summarizes the character-level perplexity values for all tokenization methods including FST segmentation only and FST segmentation with character or BPE backoff strategy for `ess` and `grn`. For all settings, FST-based segmentation resulted in the best model for the two languages. The clear difference between FST-based models and non-FST-based models suggest that the Morfessor and BPE models failed to capture the morphological information present in the data.

The fact that the FST segmentation only worked the best for `ess` might suggest that the FST segmentation for the language might have been more robust than `grn`. Indeed, the FST segmentation only resulted in high perplexity in modelling `grn` under the 2) *New Testament* setting. With the BPE and character backoff, `grn` FST models still worked the best, but it is speculated that the FST morphological segmentation alone for `grn` might not have been reliable or the coverage of the FST was not as good as the `ess` FST.

After comparing different tokenization methods per language, we compared different languages to see which language is easier or harder to model. This line of inquiry has been pursued by several recent studies (Cotterell et al., 2018; Mielke et al., 2019; Gerz et al., 2018), where various languages are modeled using a state-of-the-art neural language model to compare relative difficulty of modelling a language with particular linguistic features. It should be noted that our data per language were not parallel so the comparison has to be drawn with caution. However, we still attempted the comparison here as comparing our models may provide insights for future studies given that we used the same or very similar RNN language models as the previous literature and that polysynthetic languages have not been discussed in this line of inquiry. If we compared the character-level perplexity, Table 5.5 and Table 5.6 show that `iku` was the easiest to model under the 1) all data setting and `eng` under the 2) New Testament setting. However, character-level perplexity may not be the right metric to use to compare different languages. The problem with the character-level measure is that it does not tell us much about real-life applications, where the difficulty of predicting an entire word might be more meaningful. More importantly, the character-level perplexity underestimates the difficulty of modelling polysynthetic languages as they tend to have longer, morphologically complex words. In fact, when we look at the word-level perplexity, the differences between polysynthetic languages and others become clearer. Table 5.9 and Table 5.10 show the word-level perplexity measures for the two experimental settings. When considering the difficulty of predicting the next word in the languages than the next character, `iku` is no longer the easiest to model under any condition. The word-level measure clearly shows that `eng`, followed by `spa`, was the easiest to model. Comparisons of the word-level perplexity values suggest that `ess`, `esu` and `iku` are quite similarly hard to model while `grn` is less difficult even though it is still quite harder to model than language like `eng` and `spa`. This observation agrees with our previous observation about
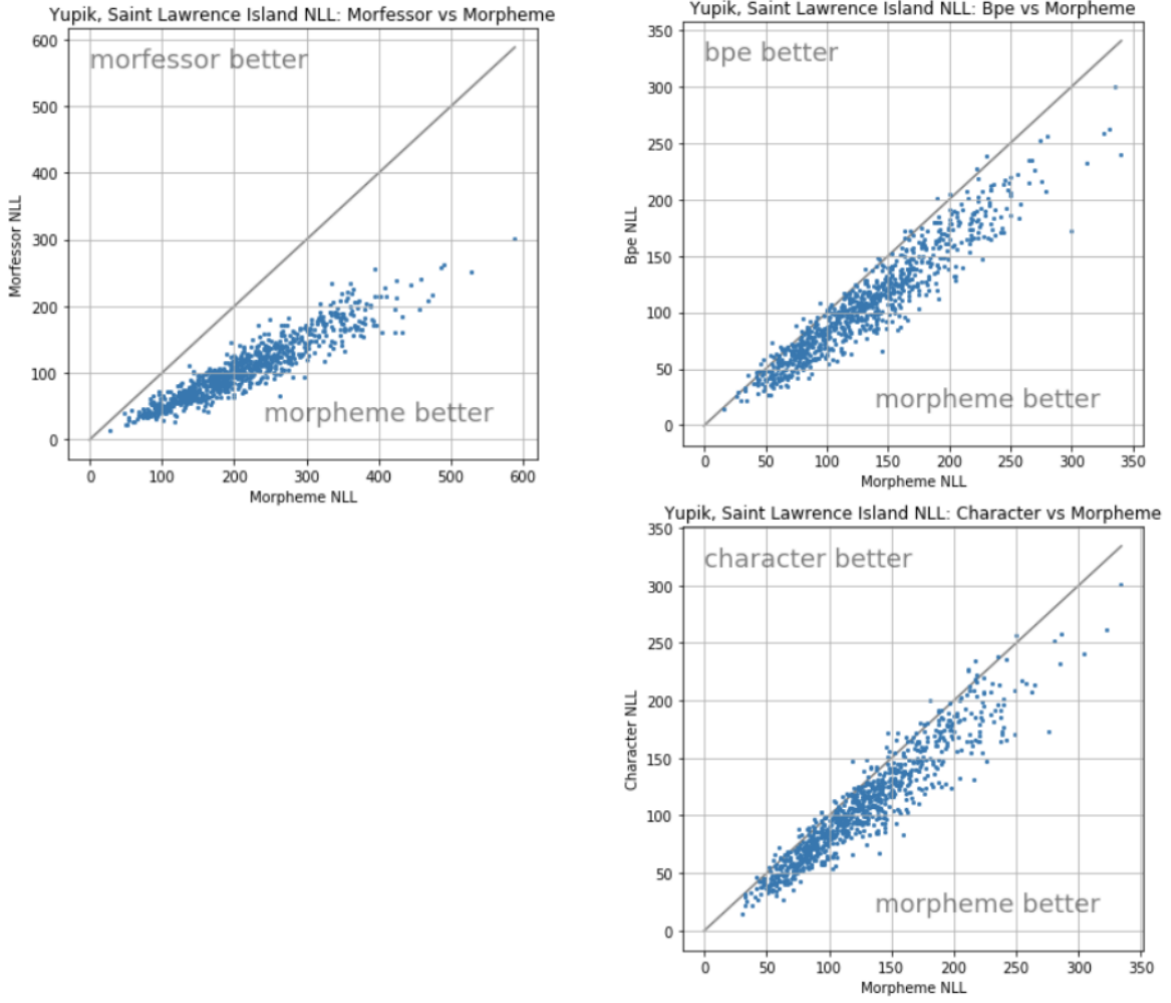
Figure 5.1: Model comparison in sentence-level negative log-likelihood for `ess`

the descriptive statistics of the datasets.

Of course, it might be unrealistic to expect that a model for polysynthetic languages would result in a word-level perplexity comparable to that for `eng` given the linguistic difference. Polysynthetic languages tend to have longer and diverse word forms because of their richer morphology. Therefore, they are likely to be harder to model than other languages. However, comparing the character-level perplexity only may result in mistakenly arguing that `iku` is easier to model than `eng`.

While the relative performance of each tokenization method for a given language stays the same regardless, the choice of the unit for the perplexity measure should be carefully made if we are to compare different languages. As mentioned above, the datasets were not strictly parallel across the languages even under the 2) *New Testament* setting. Parallel texts and different evaluation methods might facilitate comparison across languages. For example, Mielke et al. (2019) uses the average surprisal (negative log-likelihood loss) per verse when comparing languages models using data fully aligned at the verse level and also suggests a statistical method to estimate the difficulty coefficient of a language given some missing verses. Aligning a parallel corpus of polysynthetic languages and others at the verse or sentence level may lead to a more useful comparison in future research.
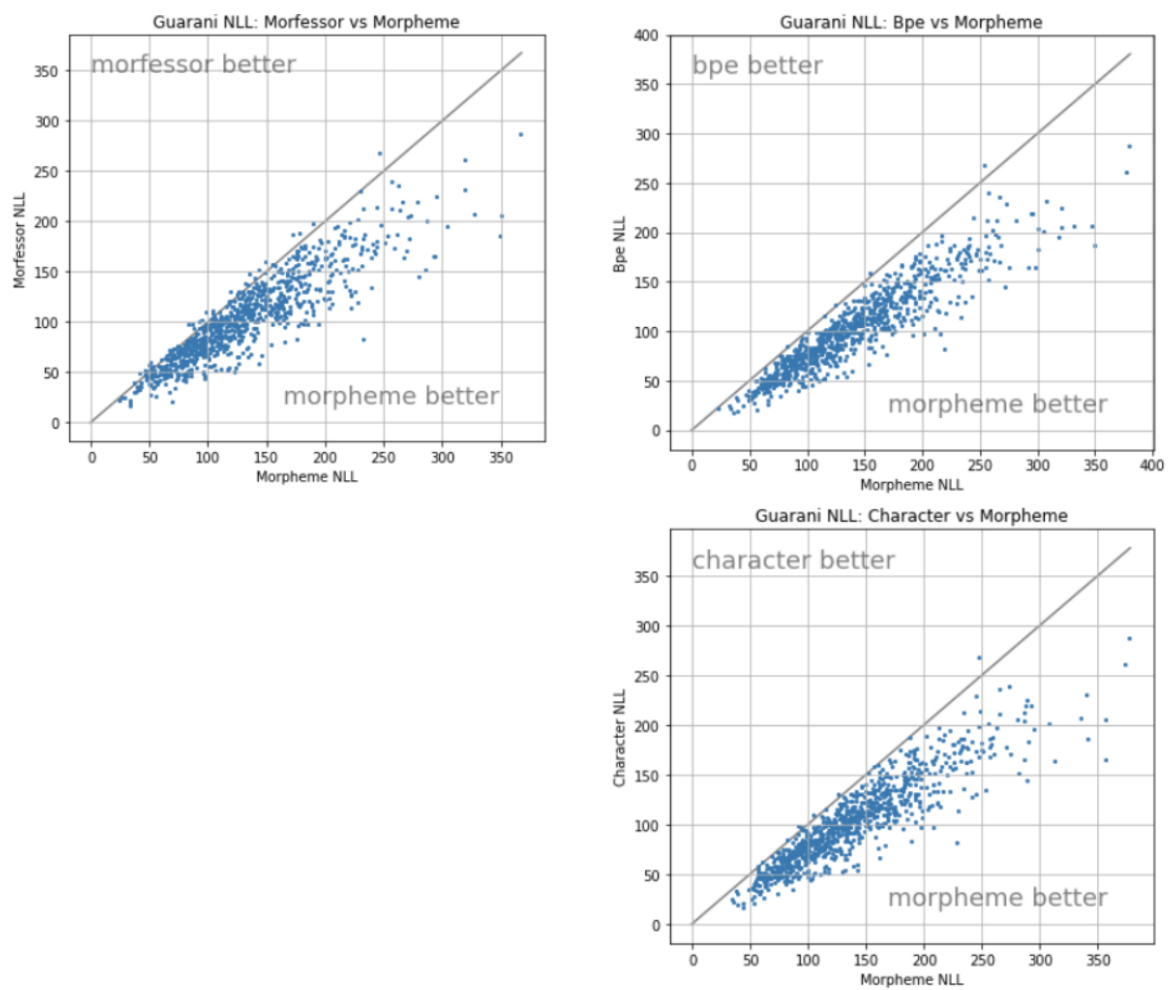
Figure 5.2: Model comparison in sentence-level negative log-likelihood for `grn`

5.6 Future Direction

The results clearly show that FST segmentation is helpful in modelling polysynthetic languages. While we had only two languages to experiment with FST segmentation, FST segmentation with or without a backoff strategy resulted in the best model by a large margin. Figure 5.1 and Figure 5.2 visualize the relative performance of the FST model v. BPE or character models at the sentence level for `ess` and `grn`, respectively. For both figures, points under the 45 degree line mean lower loss or better performance for the FST model than Morfessor, BPE or character model. For both languages, it is clear that the FST models resulted in lower loss (negative log-likelihood) per sentence overall as well as the entire text. This represents an opportunity to utilize an existing, linguistically-oriented system in aiding neural language modelling. While FSTs might not be as helpful in modelling high-resource languages with poor morphology, they will be essential in modelling low-resource polysynthetic languages.

Another line of inquiry we are currently pursuing is comparing polysynthetic languages with other languages in terms of language modelling difficulty. In order to compare different languages more precisely, we are using aligned Bible datasets and comparing a perplexity measure per verse. By modelling 149 Bibles in 94 languages, covering 24 language families, we aim to answer if polysynthetic languages are indeed harder to model than other languages and what kind of linguistic, typological features (if any) explain such difficulty.

Chapter 6

Applications & Future Work

6.1 On-device Text Prediction

One of the goals of this workshop was to make progress in providing human language technologies that can actually be used by native speakers. As smartphones become ubiquitous in native communities, text entry is becoming an increasingly important use case.

In particular, users should have access to text entry methods, namely custom keyboards, that allow them to enter text quickly and accurately. Currently, most of the languages we consider have no form of predictive keyboard available.

Our goal was to develop a pipeline for constructing custom predictive keyboards for polysynthetic languages. We wanted the keyboards to allow both automatic completion of the current unit of text being typed by the user (where units could refer to morphemes or words) and prediction of the next unit when the user input reached a boundary. Both completion and prediction rely on language models to work, so the bulk of our efforts focused on adapting trained neural network language models for on-device use.

Ultimately, we successfully built functional prototype on-device keyboards for Guaraní (`grn`) and St. Lawrence Island Yupik (`ess`). To our knowledge, these would be the first open-source predictive keyboards available for these languages on the Android platform.

6.1.1 Open Source Stack

We chose to integrate our predictive LM models with the `android` branch of the open source Divvun toolkit[1]. Divvun was chosen since it is actively developed, and the project has a stated goal of enabling text entry for low-resource languages. The toolkit provides base IME front end source code that handles on-device keyboard display and capturing of user input. We rewrote Divvun's default back end to enable loading a trained neural LM that could be used to make future predictions based on the text buffer content the user has already typed.

6.1.2 User Interface Considerations

Polysynthetic languages pose unique challenges for UI/UX design in the context of a predictive keyboard. A key question concerns the level of granularity at which predictions should be presented.

Existing keyboards almost exclusively make predictions over whole words. For polysynthetic languages, word-level prediction is problematic. For reasons introduced in Chapters 1 and 2, it isn't feasible to train an effective language model over words in languages with extremely productive morphology. Most words are composed on-the-fly, and so would not have been seen during training. Furthermore, polysynthetic morphology permits extremely long words (e.g., "oñembohuguaipu'ã" in Guaraní). The small prediction strip present on device keyboards would not be able to comfortably accommodate so many characters in a single prediction.
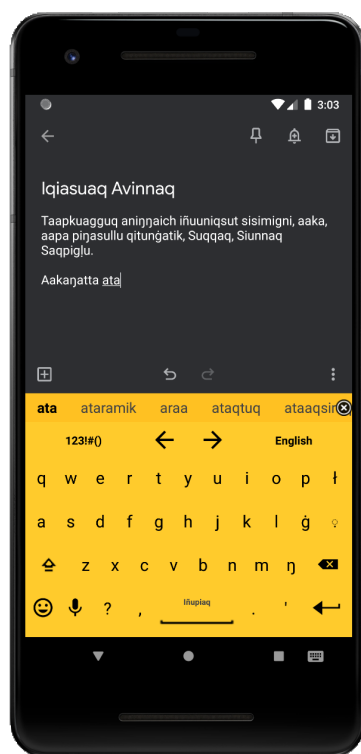
---

[1]`https://github.com/divvun/giellakbd-android`

Figure 6.1: Sample mobile keyboard interface.

As a compromise, we chose to use morphemes as the unit of prediction for our keyboard prototypes. As the user types, the prediction bar presents them with either completions of the current morpheme they are in the middle of, or predictions for the next morpheme if the language model predicts they are at a morpheme boundary.

The use of morphemes as units of prediction implies that we have access to morphological analysis and segmentation tools that can generate morpheme-level training data for our language models. These tools may not be available for all languages, in which case different subword units may need to be used. One option is do modelling and prediction over BPE word chunks. However, these would likely appear unnatural to most users, since BPE segmentation is unsupervised and linguistically unaware, leading to segmentation that doesn't correspond to any natural boundaries. A better option would be to use syllables as units, since they can be extracted with a simple model that looks for consonant/vowel alternations, and do correspond to cognitively 'natural' linguistic units.

6.1.3   Adapting Neural Language Models for Mobile Devices

As shown in Figure 6.1, we'd like to build an interface that uses the context typed into a buffer to present completions and predictions to the keyboard user. To do this, we need to feed the context data into a language model.

Initially, we attempted to use the SOTA PyTorch-based language models tested in Chapter 5 directly on-device. However, this proved to be technically prohibitive. First, device resources are limited, and keyboards should be lightweight — they only account for text entry and shouldn't have a significant impact on other running applications. We set a goal of keeping keeping our model size on the order of 10Mb. Second, there is little built-in support in Android for loading and running PyTorch models. In contrast, Google provides the TensorFlow Lite(TFLite) framework for loading models trained via TensorFlow and converted for on-device use.

We attempted to convert our PyTorch models to TensorFlow using the ONNX, toolkit[2] but found that the automatic converter did not support many of the operations used. Ultimately, we settled on training custom models for keyboard operation building on TensorFlow sample code.[3] We trained our models using the full desktop

---

[2]`https://onnx.ai`
[3]`https://www.tensorflow.org/tutorials/sequences/recurrent`

version of TensorFlow, and successfully exported the portion of the resulting computation graph responsible for inference to TFLite.

For both Guaraní and Yupik, language models were trained on text from the Bible, that had been processed via the FSTs described in Chapters 3 and 7 to include morpheme boundaries. The data was split as described in Chapter 5 for consistency with the language modelling experiments described there. The training data covered all available Bible verses except the gospel of Luke (which was reserved as development data), and John (which was reserved as test data). The models were built at the character level, but with morpheme boundaries (@) marked directly on predicted symbols, as shown in Figure 6.2. This modification enabled the model to guess when a morpheme boundary was reached (i.e., a symbol with @ was predicted/typed).[4]
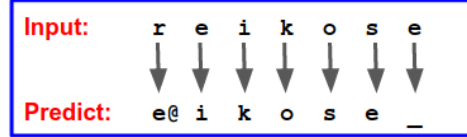

Figure 6.2: Language model training for keyboard.

The model consisted of the following architecture. A single LSTM with 2 layers, and 200 hidden units per layer, read a 30-character context. The final hidden state of the LSTM was passed through a dense layer followed by a softmax to assign probabilities to each possible next symbol. The LSTM was trained with dropout (keep_prob=0.75) between layers, with dropout disabled during inference. Batches of 20 contexts were used for training. Optimization was done via Adam, with initial learning rate 1.0 and learning rate decay 0.5.

When the model was loaded on the device, our custom Divvun back end sent the last 30 chars of the input buffer the user had typed through the model, and used the greedy algorithm below to generate continuations and predictions to display to the user in the keyboard's prediction bar.

Algorithm 2: Greedy continuation/prediction generation

Result: N prediction candidates
predictions = list;
```
/* Get the LM's ranked predictions for the next char          */
```
nextFromLM = LM.predict(context[-X:];
```
/* Loop over top N continuation points                        */
```
for *c in top N from nextFromLM* **do**
```
    /* Greedy unroll to fill out prediction candidate        */
```
    prediction = c;
    tmp = (context + c)[-X:];
    while *boundary symbol (@,_) not yet reached* **do**
        nextFromLM = LM.predict(tmp);
        c = top 1 char from nextFromLM;
        prediction += c;
        tmp = (tmp + c)[-X:];
    end
    predictions.append(prediction)
end

Currently, prediction stops when the model predicts a morpheme or word boundary. This stopping condition can be altered as needed to, for example, avoid stopping if the current prediction is too small (e.g., a single character) or continue predicting until the total log probability of the predicted string drops below a given threshold. Predictions can also be reached by a different, less greedy search algorithm, such as a depth first search starting at the current context. However, this has a high chance of producing many candidates with the same prefix. The method used here was chosen for its simplicity, and because it ensures candidates are diverse (no two can-

---

[4]Note that morpheme boundaries *never* appear in the user's input buffer according to this scheme. This is different from a system based entirely on words, as the relevant boundaries, spaces and punctuation symbols, are visible.

didates can share the same initial character). User testing might be able to determine if this bias towards diverse predictions is desirable.

6.1.4 Future Development

In Chapter 5, we evaluate our underlying language model quality via perplexity measures. Unfortunately, we did not have access to native speakers during the workshop and so could not perform direct user testing with our prototype keyboards.

Our ultimate goal would be to push our development back to the main Divvun project, so that it can receive ongoing support, and make it into the hands of native speakers. However, there *are* a number of evaluation measures that approximate the user experience related to prediction quality. Top-n prediction recall measures how often the correct prediction would have been shown to the user in the keyboard's prediction strip (assuming the user was typing a fixed script). Similarly, we can measure how many keystrokes a user can save by selecting a prediction (1 touch) versus typing it out (# touches corresponding to characters in the prediction unit).

Our prototype keyboards lack certain features that are standard on more mature offerings for languages like English. First, we assume the users touch exactly the keys they intended, and that they don't make spelling mistakes. The reality of using a touch device is that input is noisy and prone to error, with touches often sensed only in the vicinity of the intended key. A noisy channel model applied to the sequence of touch points received by the keyboard can be used to auto-correct these mistakes.

Second, our keyboard's predictions are at the mercy of the data used to train our language models. Without a whitelist of acceptable units, or a blacklist of units that shouldn't be predicted, there is nothing preventing the model from generating offensive language. Similarly, predictions can be significantly biased towards the style of the training data. In our case, the our LMs are noticeably 'evangelical,' being trained almost exclusively on text from the Bible.

6.2 Speech Recognition

Within this section, we discuss two experiments with automatic speech recognition on polysynthetic languages: preliminary experiments with Crow (cro) word prediction and experiments with Guaraní speech recognition. First, we describe previous work on speech recognition for polysynthetic languages as well as some of the inherent difficulties that arise when constructing speech corpora. Then, we discuss our baseline approach to end to end neural speech recognition using the Deepspeech model (Hannun et al., 2014), the preliminary results obtained and a discussion of future directions for polysynthetic speech recognition.

6.2.1 Related work

Speech recognition for polysynthetic languages is a relatively new area of research. Much of this is due to the necessity of large transcribed speech corpora.

Klavans et al. (2018b) presents an overview of the challenges facing automatic speech recognition for polysynthetic languages. They note that there is a dearth of resources for polysynthetic languages, particularly transcribed speech corpora. These corpora require large volumes of data from skilled native language speakers. The size of the corpora required and the linguistic, technological and language specific knowledge required make this an difficult task for communities to accomplish on their own. Hasegawa-Johnson et al. (2017b) states that "transcribing even one hour of speech may be beyond the reach of communities that lack large-scale government funding" (as cited in Klavans et al. (2018b)).

For Seneca, Jimerson et al. (2018) investigated the application of different ASR models to a small spoken corpus of Seneca (consisting of approximately 155 minutes of recordings). They found that GMM ASR models from the Kaldi ASR toolkit Povey et al. (2011) yielded better results than neural approaches on this small dataset size – requiring transfer learning from pretrained English ASR models and various augmentation procedures on both the text data and audio data to even approach GMM performance.

For Guaraní, a relatively large speech corpus has been constructed as part of the IARPA Babel project.[5] This

---

[5]Though as noted in Gales et al. (2017), the BABEL corpora are small in comparison to other corpora used in end to end neural ASR. Hannun et al. (2014), for example, used 5,000 hours of data.

Learning rate	WER	CER
10^-3	97.07	87.11
10^-4	98.97	85.76

Table 6.1: Crow speech recognition

dataset enabled the development of several existing speech recognition systems. Hartmann et al. (2016) experimented with GMM and DNN models on several of the BABEL languages including Guaraní, finding overall better performance for DNN models. Their main contribution was innovative data augmentation techniques. They sampled noise from sections of the BABEL dataset without speech data.[6] This noise was then injected into the regular transcribed data at a signal to noise ratio between 0 and 20 db. An additional data augmentation method employed by Hartmann et al. (2016) involved speed peturbation. Previous research Ko et al. (2015), found that sampling the audio signal at different rates was an effective data augmentation technique. Using these two methods in combination, Hartmann et al. (2016) see a reduction in word error rate from 46.7 to 45.2.

Gales et al. (2017) also worked with Guaraní. They use an end to end neural approach, as we do, but they leverage stimulated network training. Stimulated network training aims to train networks where nodes with similar activation properties are grouped together Gales et al. (2017). Their paper also discusses a number of optimization methods for keyword search in speech data. They obtain a WER of 49.5 for their Guaraní ASR system using stimulated network training.

6.2.2 Methodology

Deepspeech

Hannun et al. (2014) introduces the end to end neural speech recognition system used for the following experiments. This system takes in short time fourier transform (STFT) features (referred to as 'spectrogram' features in the original work). These features go through three convolutional layers with ReLU activation, and then a single bidirectional RNN. Lastly, a softmax layer is used to give a probability distribution over the possible characters in the dataset.

We borrow from this original implementation with some modifications: instead of a simple recurrent layer, we utilize gated-recurrent units, and instead of a single hidden recurrrent layer, we utilize a number of different recurrent layers. Hannun et al. (2014) use a non-gated recurrent final layer as they were seeking to avoid computing and storing the update, input and output gates used in Long-Short-Term-Memory (LSTM) recurrent units. As a compromise between LSTMs and non-gated RNNs, we utilize Gated Recurrent Units (GRUs). Gated Recurrent Units have an update gate but no output gate, thus saving some computation in comparison to an LSTM but also allowing the neural network to be less susceptible to exploding/vanishing gradients. We also introduce more recurrent layers after the convolutional layers with significant increases in performance at the cost of increased runtime.

6.2.3 Decoding

Language models can help improve automatic speech recognition systems by imposing constraints on the possible character co-occurances. We present results for greedy decoding, where no language model is utilized and the network's predicted character sequence is not explicitly constrained. In the future, we will incorporate language models into the speech recognition system.

6.2.4 Preliminary results

Initial results for Crow word recognition and Guaraní speech recognition are shown in the following sections.

---

[6]These sections are denoted as **<no-speech>** in the transcription files

Number of GRU layers	WER	CER
1 layer	92.98	52.75
2 layer	87.85	47.90
3 layer	86.00	46.96
4 layer	82.08	44.40
5 layer	82.00	44.50

Table 6.2: Guaraní results using greedy decoding

Number of GRU layers	WER	CER
1 layer	92.36	51.89
2 layer	86.44	47.18
3 layer	83.74	45.49
4 layer	82.73	44.46
5 layer	81.80	44.45

Table 6.3: Guaraní results using greedy decoding and data augmentation

6.2.5 Crow

As noted in 3.1.5, the data available for Crow consists only of recordings of single words and small phrases. In addition, very little monolingual text data for Crow was available. Due to the lack of long phrases, as with the Guaraní data, and the lack of large monolingual language resources, only a single recurrent layer was used in our model, similar to the original DeepSpeech implementation. In addition, the language model created from a very small collection of Crow monolingual stories was given very little weight due to the low coverage of the model. Initial experiments at word prediction proved unsuccessful. The neural net simply produced all spaces for output.

A pretrained English model trained on the Librispeech corpus was leveraged in an attempt to get any output at all from the Crow data. This pretrained model was then adapted to the available Crow data. The results from this adapted speech recognition model are shown in Table 6.1. While the results produced are very poor, the network was at least producing some output at this point.

6.2.6 Guaraní

For Guaraní, a number of different recurrent layers were used. Character and word error rates for the development dataset from the IARPA corpus using greedy decoding are shown in Table 6.2. Both the development and training dataset used only utterances between 1 and 15 seconds in length, thus the results shown are not directly comparable to Hartmann et al. (2016). Future experiments will be conducted on all the data for more direct comparison. All models were trained for 50 epochs with a starting learning rate of 10^-4 and learning rate annealing each epoch.

6.2.7 Future directions

Moving forward, we will incorporate neural language models into the speech recognition systems. Currently, the results displayed utilize simple greedy predictors with no explicit language modelling or conventional n-gram based language models (Heafield, 2011) for decoding. Gales et al. (2017) use an RNN language model with Pashto speech recognition and found that it had a minor effect on speech recognition but helped significantly with keyword search. However, their approach seems to involve a neural language model during the decoding stage. Incorporating a neural language model into the architecture using adversarial networks could enable still lower error rates as the model

Chapter 7

Feature-rich Open-vocabulary Interpretable Language Model

In this chapter, we present a novel general-purpose neural language modelling framework designed to be capable of handling a broad variety of typologically diverse languages, including languages whose morphology includes any or all of the following: prefixes, suffixes, infixes, circumfixes, templatic morphemes, derivational morphemes, inflectional morphemes, and clitics. In this chapter we motivate our language modelling framework using examples drawn primarily from St. Lawrence Island Yupik. St. Lawrence Island Yupik is a polysynthetic suffixing language in which words with 1 root, 0–3 derivational morphemes, and 1 inflectional are common, and words with up to 7 derivational morphemes have been attested (de Reuse, 1994).

(4) Qikmighhaak   neghtuk
     qikmigh -ghhagh -k   negh   -tuk
     dog   -small   -ABS.DU   to.eat -INTR.IND.3DU

     'The two small dogs eat'

In Example (4) we observe a sample two-word sentence from St. Lawrence Island Yupik. The first word *qikmighhaak* is a noun composed of a noun root *qikmigh*, a derivational suffix *-ghhagh* that serves as a diminutive, and an inflectional suffix *-k* that indicates the noun's case (absolutive) and number (dual). The second word *neghtuk* is a verb composed of a verb root *negh* and an inflectional suffix *-tuk* that indicates the verb's mood (indicative) and valence (intransitive), as well as the person (3rd person) and number (dual) of the verb's subject. Note that it is common for the form in which a morpheme surfaces in a word to differ from the underlying lexical form of that morpheme. In the morphemes' respective surface forms in this example, the final uvular fricative of *qikmigh* and *-ghhagh* are each dropped, the vowel of *-ghhagh* is lengthened, and the final uvular fricative of *negh* devoices to match the adjacent voiceless stop at the beginning of *-tuk*.

(5) Mangteghaghrugllangllaghyunghitunga
     mangteghagh- -ghrugllag- -ngllagh- -yug-   -nghite- -tu-   -nga
     house-   -big-   -build-   -want.to- -to.not- -INTR.IND-   -1SG
     'I didn't want to make a huge house'   (Jacobson, 2001, pg. 43)

In Example (5), a single Yupik word represents an entire sentence. The word consists of a noun root *mangteghagh*, a derivational suffix *ghrugllag* that serves as an augmentative, a verbalizing derivational suffix *ngllagh*, a verb-elaborating derivational suffix *yug*, another verb-elaborating derivational suffix *nghite*, and inflectional suffixes *tu* and *nga* that mark mood (indicative) and valence (intransitive), as well as the person (1st person) and number (singular) of the verb's subject.

7.1 Language Model Desiderata

A language model capable of effectively modelling the full linguistic diversity found in human languages, including St. Lawrence Island Yupik and similar endangered and polysynthetic languages, should have the following desiderata.

7.1.1 Flexibility with respect to language typology

Typical methods of categorizing languages by morphological type include isolating, fusional, agglutinative and polysynthetic. There are also morphological affix types such as prefixes, suffixes, circumfixes, infixes and templatic morphology, and processes such as compounding and incorporation.

One can think of isolating languages as those (almost) without productive morphology, such as Chinese and Vietnamese. These languages are well served by existing approaches to language modelling which treat the word as the fundamental unit.

Fusional languages are those where a morpheme may represent multiple morphological or syntactic features. Most well-known Indo-European languages are of this type. They may also have complicated, irregular, or lexicalised phonological processes occurring when morphemes are joined together. Consider for example Catalan *tener* 'to have'—*tinc* 'I have'—*tinga* 'I have'. The stem is *ten-*, *-er* is the formant of the infinitive, *-c* is the formant of the first person singular present indicative and *-nga* is the formant of the first and third person present subjective. A vowel change in the stem occurs when the suffixes are attached to the stem. This example has two fusional features: multiple features per morpheme and stem-internal phonological changes caused by affixing. These languages are fairly well dealt with in existing approaches, the number of forms that can be generated by these processes may be larger than in isolating languages, but is essentially a finite-set.

As mentioned, current *ad hoc* methods work fairly well with isolating and fusional languages, where there are a finite number of forms for a single word. Out of vocabulary items are a problem, but are typically related to unseen new stems rather than forms of seen stems. Agglutinating and polysynthetic languages have this problem too, but in addition they have the problem of unseen forms of previously seen stems.

In agglutinating languages — and in polysynthetic languages to an even greater extent — words are typically made up of many morphemes concatenated together. These are typically with prefixes or suffixes, or a combination. The Yupik example in (4) is an example of suffixing, and indeed Yupik is an exclusively suffixing language. Guaraní combines suffixes, which are primarily for tense, aspect, and mood (TAM) markers and subordination, with prefixes for valency changing and agreement. This is illustrated in Example (6) where the *ai-* prefix indicates first-person singular agreement, and the *-se* suffix indicates volitional mood, and in Example (7) where the *ña-* prefix indicates agreement and the *-va* suffix indicates nominalisation.

(6) Aikosénte
     Ai-ko-se-nte
     SG1-live-VOL-JUST
     'I would just like to live'

(7) ñaha'arõ'ỹetéva
     ña-ha'arõ-'ỹ-ete-va
     PL1-wait-NEG-INTS-REL
     'that we did not expect at all'

The negative form of Guaraní verbs is formed by a circumfix of two morphemes, *nd-* and *-i*. These circumfixes go around verbal derivations, agreement and (TAM) markers etc, as in (7.1.1).

(8) ndojuhumo'ãi
     nd-o-juhu-mo'ã-i
     NEG-3-find-FUT-NEG

In Chukchi the comitative case is made up of a circumfix of two morphemes, /ɣa/- and -/ma/. The noun /ɬawt/

'head' forms the associative singular /ɣaławtăma/ by combining these and adding an epenthetic schwa.

Infixes are morphemes that break a given stem and appear inside it. For example in Seri, a language spoken in the north-west of Mexico. It uses infixation after the first vowel in the root to create forms with number agreement. For example, *ic* 'to plant', *i{tí}c* i 'did she plant it?' vs. *i{tí}{tóo}c* 'did they plant it?'.

In languages with templatic morphology, the root is typically represented as a consonant template, e.g. in Maltese, *k-t-b* 'book'. Inflection takes place by "filling" the slots in the root with other templates, such that e.g. *ktieb* 'book' (singular), *kotba* 'books', are formed by combining the root with the vowel templates {ø-ie, o-ø}, and in the plural the suffix *-a*.

An ideal language model would be able to encode all of these types of morphology in a generic and compositional manner without using language- or typology-specific tricks or assumptions (e.g. productive morphological processes are exclusively suffixing).[1]

It should allow for arbitrary subsets of characters in a given string to form meaningful, compositional units.

7.1.2 Ability to incorporate external knowledge sources as features

In high-resource settings, neural networks commonly function as effective feature extractors (Goodfellow et al., 2016). In very low-resource settings such as St. Lawrence Island Yupik, extreme data sparsity means that neural models are likely to have insufficient data to effectively extract such reliable features. To alleviate this issue, our language model should be capable of incorporating a rich array of features from supplementary knowledge sources when insufficient data conditions prevent learning them.

Finite-state morphological analyzers (Beesley and Karttunen, 2003) in particular represent a mature technology capable of serving as a reliable source of rich linguistic features. In the Yupik Example (4) above, we make use of the finite-state morphological analyzer of Chen and Schwartz (2018). At a minimum, we expect such an analyzer to decompose a Yupik word, providing morpheme boundary information and the associated constituent morphemes. We expect that in most cases a morphological analyzer should also provide the underlying orthographic form of each root morpheme and each derivational morpheme, the set of linguistic features such as noun case, verb mood, person, and number associated with each inflectional morpheme, and the underlying type of each morpheme (such as noun, verb, nominalizing suffix, etc). In the some cases, an analyzer might also provide information regarding the phonemes in each morpheme.

7.1.3 Open vocabulary

In high-resource languages, especially those that are analytic rather than synthetic, a common approach is to treat morphologically-distinct variants (such as *dog* and *dogs*) as completely independent word types, rather than inflected variants of a common root. In polysynthetic languages in general, and in Yupik in particular, encountering previously unseen word forms is pervasive and should be considered the norm rather than the exception. In very low-resource settings, it is especially important that our language model be able to robustly handle and predict out-of-vocabulary tokens. Language models with a closed vocabulary are not viable in such settings. Instead, we require an open vocabulary language model in which the probability of a token given a history can be robustly calculated even when that token was not present in the training data.

7.1.4 Interpretability of predicted units

By definition, a language model provides a probabilistic model over a sequence of linguistic units. In other words, a language model must be able to provide a probability distribution over the identity of the current linguistic unit given a history representing the preceding linguistic units in the sequence. We use the term linguistic unit to refer to an instance of any well-defined linguistic level of analysis, such as a word, a morpheme, a syllable, a phoneme, or even a grapheme.

In our language model, we require that the computational mechanism that implements the linguistic unit be interpretable. For example, consider the case of a trained instance of our language model randomly generating a sequence of morphemes; when the model generates a morpheme, we should be able to recover whatever rich

---

[1]We would note that treating words as basic units can also be considered to be a language-specific trick designed for isolating and fusional languages.

features may be encoded therein (see §7.1.2), such as the underlying grapheme or phoneme sequence and the type of morpheme (root, derivational, inflectional, etc). This should be the case regardless of whether the generated unit was present in the training data or not (see §7.1.3).

7.2   Sub-word language models

The rich morphology and phonology of Yupik and typologically similar languages results in an extreme type-token ratio. This fact coupled with a very small corpus size make the use of n-gram language models and recurrent neural language models over words highly unlikely to be effective. Schwartz et al. (2019) examined the number of potential word forms word forms in St. Lawrence Island Yupik, and estimated approximately 1.27×10^{23} morphotactically licensed word forms. This number is approximately equal to current estimates of the number of stars in the observable universe.[2] While this estimate does not take into account restrictions imposed by semantic felicity, the polysynthetic nature of the language ensures an extremely high fraction of *hapax legomenon* in Yupik texts, with Schwartz et al. (2020) reporting that approximately every other Yupik word token establishes a previously unseen word type. In contrast to the astronomical number of potential Yupik word forms, the complete collection of fully digitized St. Lawrence Island Yupik texts available at the time of the 2019 JSALT workshop consisted of a corpus of slightly over 81,000 word tokens (see Chapter 3 for more details). In lieu of word-based language models, we consider language models that utilize sub-word units.

Language models serve as an enabling technology for other downstream language technologies, including mobile text prediction. These technologies are mature and widespread for many high-resource languages, but relatively immature and rare for polysynthetic languages. In this section, we present several motivating use cases of sub-word language models for polysynthetic language.

7.2.1   Prediction of next morpheme

The core operation of a language model is estimating the conditional probability of a predicted next linguistic unit given a history of previous linguistic units. Figure 7.1 illustrates a recurrent neural network language model that predicts the most likely next morpheme given a history of four immediately preceding morphemes, where each morpheme is encoded as a vector.

Figure 7.1: A recurrent neural network language model over morphemes can be used to predict the next morpheme in a sequence. In this figure, the light green boxes represent Yupik morphemes from Example (4), each encoded as a vector.

7.2.2   Prediction of next character

A closely related task applicable in the context of mobile text completion is the prediction of the next character given a preceding sequence of characters. In the polysynthetic language setting, it may be beneficial to augment such a model with a history of morphemes in situations where this information is available.

7.3   Neural morphological analysis

As discussed in §2.1, finite-state morphological analyzers provide a mechanism for encoding linguistic knowledge in a finite-state transducer capable of analyzing a word and providing morpheme boundaries and other linguistically salient information about the underlying morphemes that comprise the word. Recent work has explored how a finite-state morphological analyzer can be used to bootstrap a neural morphological analyzer (Micher, 2018b; Schwartz et al., 2019; Silfverberg and Tyers, 2019). Building on that work, we propose a neural morphological
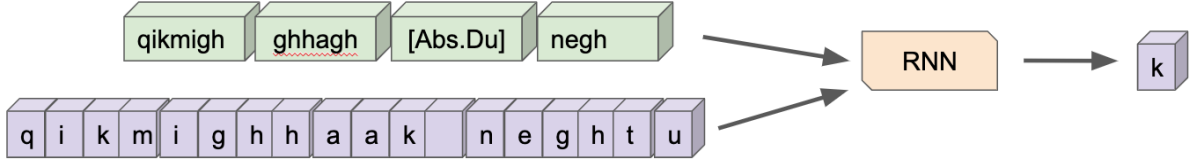
---

[2]https://www.skyandtelescope.com/astronomy-resources/how-many-stars-are-there

Figure 7.2: In a text completion setting, a more sophisticated recurrent neural network language model could predict the next character given a history of preceding characters and a history of preceding morphemes from Example (4). In this figure, the light green boxes represent Yupik morphemes while the light purple boxes represent characters.

analyzer that directly predicts morpheme vectors, rather than predicting a sequence of strings representing an analyzed form.
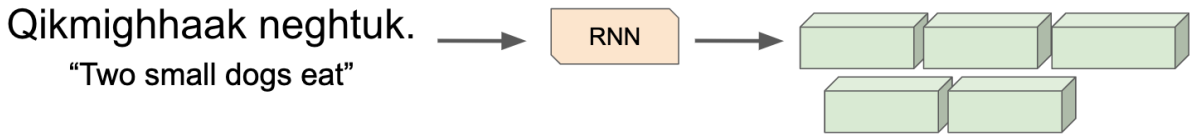

Figure 7.3: In a morphological analysis setting, a sequence-to-sequence model predicts a sequence of morphemes from an input sequence of Yupik characters from Example (4). In this figure, the light green boxes represent predicted Yupik morpheme vectors.

7.4 Tensor Product Representation

To satisfy the language model desiderata specified in §7.1, we consider the Tensor Product Representation (TPR) proposed by Smolensky (1990). The use of TPRs provides a principled way of representing hierarchical symbolic information in vector spaces, such as those used as the input and output domains of neural networks. Developing a tensor-product-based representational scheme begins by decomposing a symbolic structure into roles and fillers. A symbolic structure can then be represented as the *bindings* of fillers to roles. Once decomposed, both roles and fillers are embedded into a vector space such that all roles are linearly independent from one another. Let b be a list of ordered pairs (i, j) representing filler i (with embedding vector $\hat{\mathbf{f}}_i$) being bound to role j (with embedding vector $\hat{\mathbf{r}}_j$). The *tensor product representation* $\mathbf{T}$ of the information is then given by

$$\mathbf{T} = \sum_{(i,j) \in b} \hat{\mathbf{f}}_i \otimes \hat{\mathbf{r}}_j \in \mathbb{R}^d \otimes \mathbb{R}^n. \tag{7.1}$$

This TPR may itself be used as a filler and subsequently be bound to another role vector. This process results in a TPR that represents hierarchical compositional structure.

7.4.1 Unbinding

TPRs are useful because they embed arbitrary symbolic structure in a vector space in such a way that simple linear algebra operations may be used to retrieve the form of the symbolic structure, including its compositional structure. The core operation in retrieving this structure is called *unbinding*. We may use unbinding to query a role for its filler. Unbinding may be accomplished by any of several exact or approximate strategies. Exact unbinding requires linear independence of the roles; however, recent (unpublished) work points to the accuracy of approximate unbinding even in densely packed TPRs. In this work, we use self-addressing unbinding, as it is quick to compute and proved sufficiently accurate for our purposes. Self-addressing unbinding retrieves the filler $\hat{\mathbf{f}}_i$ for the role $\hat{\mathbf{r}}_i$ by simply computing the inner product between the role vector and the TPR:

$$\tilde{\mathbf{f}}_i = \mathbf{T} \cdot \hat{\mathbf{r}}_i \tag{7.2}$$
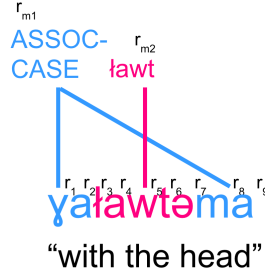
Figure 7.4: This sample word from Chukchi is composed of a root morpheme *ɫawtǎ* and a circumfix *γa...ma*. The individual characters positions in the word comprise roles r_1 through r_9, while the characters at those respective positions comprise fillers f_1 through f_9. Roles r_{m_1} and r_{m_2} represent morpheme positions within the word, and are respectively filled by f_{m_1} (denoting the identity of the circumfix morpheme marking associative case) and f_{m_2} (denoting the identity of the root morpheme).

This unbinding is exact if the role vectors are orthogonal to one another. Otherwise, the intrusion of the filler of role j, $\hat{\mathbf{f}}_j$, into the unbound filler of the role i, $\tilde{\mathbf{f}}_i$, is given by

In our case, since we have a fixed filler vocabulary, we were able to snap our unbindings to the filler with the highest cosine similarity to the unbound vector with sufficient accuracy to render this intrusion irrelevant. Other unbinding strategies involve computing an inverse or pseudoinverse of a matrix of role vectors to perform a change of basis and decrease the intrusion.

7.5   Morpheme vector representations from TPRs

We use TPRs (§7.4) to bridge the gap between the rich hierarchical symbolic information encoded in finite state morphological transducers (such as Chen and Schwartz, 2018) and the morpheme vectors needed by the neural models described in §7.2 and §7.3.

7.5.1   Morpheme TPRs

Given a language, a corpus of text in that language, and a finite-state morphological analyzer for that language, we can use the finite-state analyzer to obtain a morphological analysis for each word in the corpus. For each morpheme provided in an analysis, we extract a collection b of linguistically salient feature-value ordered pairs (i, j). Each linguistic feature j serves as a TPR role; each value i serves as a TPR filler. For each such feature j (such as noun case), we define $\hat{\mathbf{r}}_j$ to be a role vector representing that feature; for each value i (such as ABS) associated with feature j, we define $\hat{\mathbf{f}}_i$ to be a filler vector representing that value. This use of TPRs enables us to jointly encode latent structural information provided by a finite state transducer with surface information in a principled manner. This process is depicted in Figure 7.5.

7.5.2   Learning morpheme vectors using an autoencoder

The morpheme tensors constructed in §7.5.1 are potentially very high dimensional. Depending on how much linguistic information is encoded in each tensor, the morpheme tensors may consist of approximately 10^3 to 10^9 floating point values per tensor. Tensors of this size are far too large to be directly usable as morpheme representations in the neural models described in §7.2 and §7.3. To learn lower dimensional morpheme vectors, we make use of an autoencoder. The autoencoder is trained using the dictionary of previously constructed morpheme tensors. The trained autoencoder can be used to encode a low-dimensional morpheme vector from a high-dimensional morpheme tensor by running the morpheme tensor through the first half of the autoencoder, and can be used to obtain a high-dimensional morpheme tensor from a morpheme vector by running the morpheme vector though the latter half of the autoencoder.
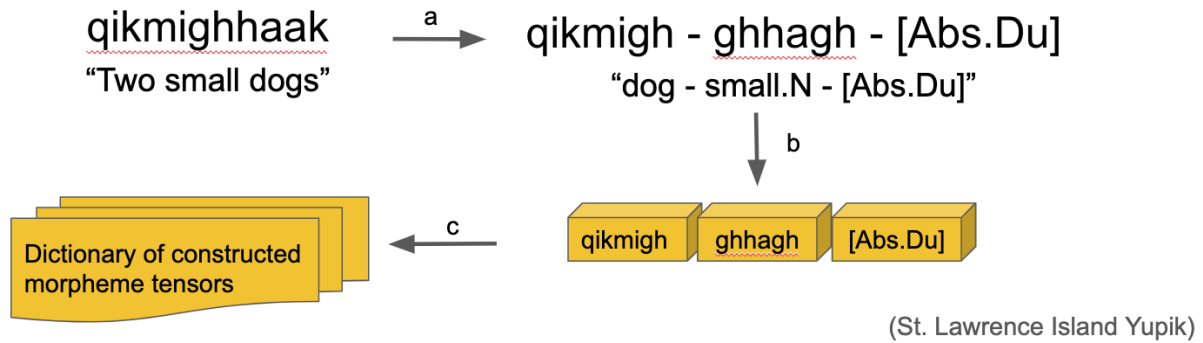
Figure 7.5: (a) Each word in a corpus is processed by a morphological analyzer. (b) A tensor product representation of each morpheme is calculated, resulting in one tensor per morpheme. (c) The morpheme tensors extracted from the corpus are stored in a dictionary.
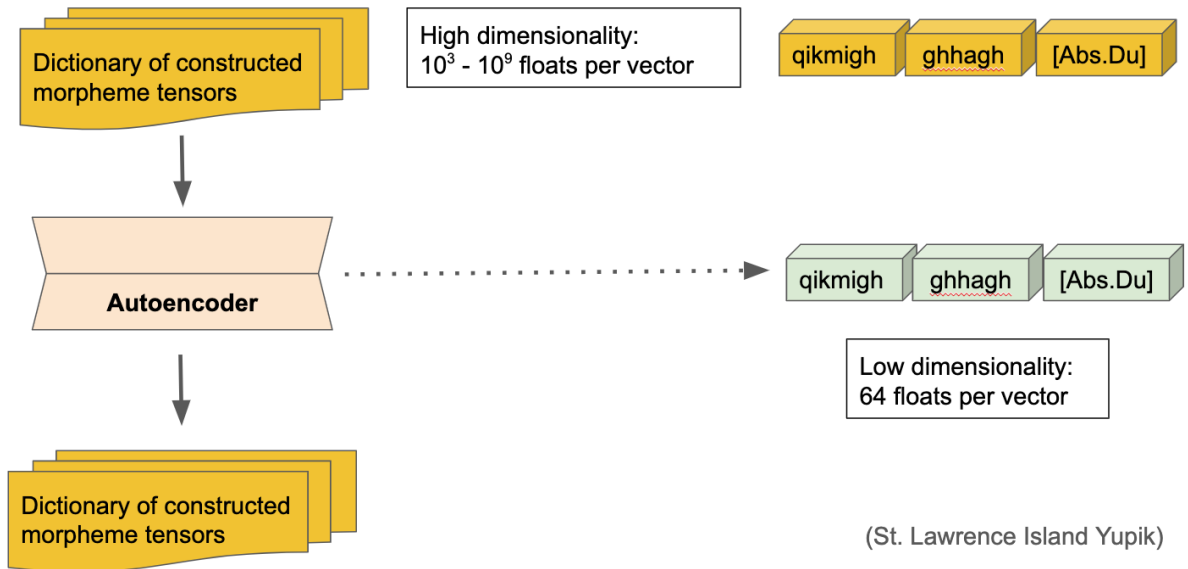

Figure 7.6: An autoencoder trained is on the dictionary of morpheme tensors.

7.6 Unbinding loss

In order to effectively train the autoencoder in §7.5.2, gold standard morpheme tensors must be compared against predicted morpheme tensors outputted by the autoencoder. However, the morpheme tensors are very high dimensional. In initial experiments, we used mean squared error as a loss function, but we found this was unable to converge for auto-encoding sparse TPRs.

To enable effective training of the autoencoder, we therefore define a novel loss function that makes use of the information encoded in the TPR. We define a loss function called *unbinding loss* that examines the unbinding properties of a predicted morpheme tensor to answer the question, "What filler is closest to the unbinding of each role in the TPR?" For simplicity, we assume the use of self-addressing unbinding in this section (which we also used in the work presented here), but the computations are analogous with other unbinding strategies, relying only on a fixed role and filler vocabulary and a fixed number of bindings. We call the output TPR **T**.

Given a predicted tensor, the first step to computing the unbinding loss is recursively unbind roles until the leaves of the structure are reached – that is, unbind each role until the result of unbinding is a single vector (rather than a higher-dimensional tensor). When this point is reached, we compute the cosine similarity between the result of unbinding and all the fillers in the vocabulary. For example, assume a depth-3 structure is encoded in a TPR, where the fillers are character embeddings, the second level is left-to-right positional roles, and the highest level

is morpheme identity. If we want to see what is bound to the first position of the English *cat* morpheme in T, we would first unbind from $\mathbf{T}$ as follows (assuming self-addressing unbinding):

$$\mathbf{f}_{cat,1} = \mathbf{T} \cdot \hat{\mathbf{r}}_{cat} \cdot \hat{\mathbf{r}}_1 \tag{7.3}$$

We then get the vector of similarities $\hat{\mathbf{s}}_{cat,1}$ between this filler and the each of character embedding vectors in the vocabulary matrix V as follows:

$$\hat{\mathbf{s}}_{cat,1} = \frac{\mathbf{f}_{cat,1} \cdot \mathbf{V}}{||\mathbf{f}_{cat,1}||\mathbf{V}^i\mathbf{V}^i} \tag{7.4}$$

where $\mathbf{V}^i\mathbf{V}^i$ denotes the column-wise vector norm of the vocabulary matrix (using Einstein summation notation).

This similarity vector can be used to define a probability distribution over possible fillers through the use of a softmax. We take the logarithm of the result of this computation to obtain log-probabilities. We call this distribution P.

$$P = \log \left(\frac{e^{\hat{\mathbf{s}}_{cat,1}}}{\sum e^{\hat{\mathbf{s}}_{cat,1}}} \right) \tag{7.5}$$

We then treat each filler vocabulary word (in this case, each character) as a class, and compute the negative log-likelihood loss over this probability distribution. The resulting loss for the first character of cat being c is then

$$loss(\hat{\mathbf{s}}_{cat,1}, c) = -\hat{\mathbf{s}}_{cat,1,c} + \log(\sum_j e^{\hat{\mathbf{s}}_{cat,1,j}}). \tag{7.6}$$

In this example, we focus on the loss for a single filler; however, as we consider tree-structured representations, the number of fillers needing to be checked is exponential with the depth of our representation. In practice, we were able to overcome this difficulty by parallelizing the independent matrix computations for the loss of all the position roles for a given morpheme, trading space for time. For more complex TPRs, a potential avenue would be to exploit the fact that most roles will be empty (and their unbindings thus a matrix of zeros) by replacing the loss computations for unbound roles with mean squared error (which need only push that part of the representation to 0).

Chapter 8

Conclusions

In motivating this JSALT workshop on neural polysynthetic language modelling, we observed the following major assumptions (usually unstated) that are pervasive in most computational linguistics and natural language processing research:

- If a technique works well on English, the technique is likely to be "language agnostic" and is likely to work well on a large variety of other languages. Various other high-resource languages such as Spanish, French, German, or Chinese are sometimes used in place of English.

- For any given word stem, there will be a relatively small number of morphological variants of that stem.

- Most or all of the morphological variants of any given word stem will appear in a sufficiently large corpus to enable learning of robust statistics.

Our work was built around explicitly challenging all of these assumptions, using a variety of polysynthetic languages and a variety of natural language tasks. The polysynthetic languages that we chose to work with present numerous significant challenges. These languages are typologically very different from English and other widely-used high-resource languages. There is pervasive use of derivational and inflectional morphology. For most word stems, there are very large numbers of potential morphological variants, very few of which occur in any given corpus. For all of the selected languages (with the exception of Inuktitut), the corpus sizes are very small (less than $60,000$ sentences).

8.1 Contribution 1: Resources

One contributing factor to the dearth of prior work on computational research on endangered polysynthetic languages is the lack of easily available corpus resources. Nearly all endangered languages are very low resource. Most CL and NLP researchers do not have the personal connections with members of endangered language communities that are often critical for obtaining data for use in research. In preparation for this workshop, our team gathered together text and speech data from various sources for a variety of polysynthetic languages. In cases where we have connections with indigenous community stakeholders and rights-holders, we have begun the process of discussions regarding community desires and possibilities for data distribution. For data that we have obtained permission to distribute, we have initiated a process of public data hosting.

8.2 Contribution 2: Machine Translation

The main contributions of our machine translation work during this workshop are as follows. With first access to the beta version 3.0 of the Nunavut Hansard (Joanis et al., 2020), we were able to provide feedback and best practices for preprocessing the dataset and shared knowledge about existing character and spelling variations in the dataset. This work contributed to the data release and publication of Joanis et al. (2020); that data is now being used in the Fifth Conference on Machine Translation (WMT20) Inuktitut-English news translation shared task.

Our work at the time constituted state-of-the-art performance on translation between Inuktitut and English. It has since been surpassed by Joanis et al. (2020), and we anticipate future improvements through the WMT20 shared task.

We collected empirical evidence on several well-known but unresolved challenges, such as best practices in token segmentation for MT into and out of polysynthetic languages, as well as an examination of how to evaluate MT into polysynthetic languages. We successfully used multilingual neural machine translation methods to improve translation quality into low-resource languages (St. Lawrence Island Yupik and Central Alaskan Yup'ik) using data from related languages (Inuktitut). Notably, our "low-resource" languages were lower resource than much of the literature, and we produced improvements without the use of large monolingual corpora (which are unavailable for these languages and many other languages of interest). We observed these improvements across both n-gram-oriented and semantic-oriented metrics.

There remain a number of open challenges in this space. We encourage caution in interpreting the automatic quality metrics, as we do not yet have human judgments of translation quality for the languages examined; human judgements from the WMT20 shared task may prove particularly valuable. Our initial results, using fairly conventional methods, for both multilingual and bilingual machine translation show promise, but we expect that there remains much room for improvement.

8.3   Contribution 3: Language Models

To our best knowledge, this paper represents the first attempt at modeling polysynthetic languages using a state-of-the-art RNN model and comparing their language modeling difficulty with that of other languages. We conduct language modeling experiments on four low-resource, polysynthetic languages (St. Lawrence Island Yupik, Central Alaskan Yup'ik, Inuktitut, Guaraní) and two high-resource, morphologically poor languages (English, Spanish), using four different segmentation methods: character, BPE, Morfessor and FST. By comparing the perplexity measure at the character level, we show that the FST segmentation method worked the best for polysynthetic languages when it was available. While the Morfessor segmentation method might improve language modeling performance for some polysynthetic languages, all the other segmentation method we considered—character, BPE and Morfessor—failed to capture the rich morphology of polysynthetic languages better than the FST segmentation that is based on linguistic knowledge of the languages. We also compared the perplexity measure at the word level to illustrate how significantly difficult it is to model polysynthetic languages.

All in all, this presents an exciting starting point for a line of inquiries into modeling polysynthetic languages and utilizing the linguistic knowledge realized in FST in modeling such languages that are morphological rich and low resource. At the same time, we invite future research into linguistic characteristics that contribute to language modeling difficulty as we continue to investigate the effect of morphological complexity in our ongoing study.

8.4   Contribution 4: Mobile & Speech Applications

As smartphones become ubiquitous in native communities, facilitating native-language communication through better technology will become an important aspect of language conservation and revitalization efforts. Building on freely available open source tools, we developed a pipeline for training neural language models that can run on-device, and loading them as a predictive back-end for on-device keyboards. This effort lead to working keyboard prototypes for Guaraní (`grn`) and St. Lawrence Island Yupik (`grn`) — the first ever input methods for these language varieties to include intelligent next-unit prediction and completion. Building the prototypes highlighted the unique requirements posed by polysynthetic languages. Their complex, productive morphology results in very long words, many of which would never appear in the training data available for language modeling, and which would be unwieldy to show to keyboard users as prediction candidates. We dealt with these problems by training character-level models that were aware of morpheme boundaries, and using morphemes rather than words as units of prediction.

The low-resource nature of most polysynthetic languages is particularly poignant for automatic speech recognition. While transfer learning can help alleviate some of the issues with data poverty, neural approaches to ASR are still not sufficient to enable usable systems.

8.5 Contribution 5: Model Development

In this workshop we proposed a novel framework for language modelling that combines knowledge representations from finite-state morphological analyzers with Tensor Product Representations (Smolensky, 1990) in order to enable successful neural language models capable of handling the full linguistic variety of typologically variant languages. To support this framework, we also defined and implemented a novel loss function called *unbinding loss* that enables gold standard morpheme tensors to be compared against predicted morpheme tensors. We implemented a prototype TPR framework that we are continuing development of as part of ongoing future work.

Bibliography

Vasilisa Andriyanets and Francis Tyers. A prototype finite-state morphological analyser for Chukchi. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 31–40, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-4804`.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 1: Gambell. Bering Strait School District, Unalakleet, Alaska, 1985. URL `http://www.uaf.edu/anla/collections/search/resultDetail.xml?id=SY980AWT1985`.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 2: Savoonga. Bering Strait School District, Unalakleet, Alaska, 1987. URL `http://www.uaf.edu/anla/collections/search/resultDetail.xml?id=SY980AWT1985`.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 3: Southwest Cape. Bering Strait School District, Unalakleet, Alaska, 1989. URL `http://www.uaf.edu/anla/collections/search/resultDetail.xml?id=SY980AWT1985`.

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. *Kallagneghet / Drumbeats*. Bering Strait School District, Unalakleet, Alaska, 1993. URL `http://www.uaf.edu/anla/item.xml?id=SY990AUKT1993`.

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. *Akiingqwaghneghet / Echoes*. Bering Strait School District, Unalakleet, Alaska, 1994. URL `http://www.uaf.edu/anla/item.xml?id=SY990AUKT1994`.

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. *Suluwet / Whisperings*. Bering Strait School District, Unalakleet, Alaska, 1995. URL `http://www.uaf.edu/anla/item.xml?id=SY900AUKT1995`.

Linda Womkon Badten, Vera Oovi Kaneshiro, Marie Oovi, and Christopher Koonooka. *St. Lawrence Island / Siberian Yupik Eskimo Dictionary*. Alaska Native Language Center, University of Alaska Fairbanks, 2008.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015. URL `https://arxiv.org/pdf/1409.0473v6.pdf`.

Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. CSLI Studies in Computational Linguistics. CSLI Publications, Stanford, California, 2003.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Proceedings of Neural Information Processing Systems*, pages 932–938, 2000. URL `http://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf`.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003. URL `http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf`.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W15-3001`.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W16/W16-2301`.

Emily Chen and Lane Schwartz. A morphological analyzer for St. Lawrence Island / Central Siberian Yupik. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC 2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1416.pdf`.

Kenneth W. Church and William A. Gale. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, 5:19–54, 1991.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2085. URL `https://www.aclweb.org/anthology/N18-2085`.

Willem J. de Reuse. *Siberian Yupik Eskimo — The Language and Its Contacts with Chukchi*. Studies in Indigenous Languages of the Americas. University of Utah Press, Salt Lake City, Utah, 1994.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Benoît Farley. Nunavut Hansard Inuktitut–English Parallel Corpus version 2.0. http://www.inuktitutcomputing.ca/NunavutHansard/info.php, 2008.

Benoît Farley. The Uqailaut Project, 2009. URL `http://www.inuktitutcomputing.ca/Uqailaut/info.php`.

Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788. URL `http://dl.acm.org/citation.cfm?id=177910.177914`.

Mark J. F. Gales, Kate M. Knill, and Anton Ragni. Low-resource speech recognition and keyword-spotting. In Alexey Karpov, Rodmonga Potapova, and Iosif Mporas, editors, *Speech and Computer*, pages 3–19, Cham, 2017. Springer International Publishing. ISBN 978-3-319-66429-3.

Michael Gasser. Mainumby: un ayudante para la traducción Castellano-Guaraní. In *Tercer Seminario Internacional sobre Traducción, Terminología y Lenguas Minorizadas*, 2018.

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1029. URL `https://www.aclweb.org/anthology/D18-1029`.

Irving J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3–4):237–264, 1953.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1032. URL `https://www.aclweb.org/anthology/N18-1032`.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *13th International Workshop on Spoken Language Translation (IWSLT 2016)*, 2016. URL `http://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_5.pdf`.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

William Hartmann, Tim Ng, Roger Hsiao, Stavros Tsakalidis, and Richard M Schwartz. Two-stage data augmentation for low-resourced speech recognition. In *Interspeech*, pages 2378–2382, 2016.

Mark Hasegawa-Johnson, Mohamed Elmahdy, and Eiman Mustafawi. Arabic speech and language technology. In Elabbas Benmamoun and Reem Bassiouney, editors, *Routledge Handbook of Arabic Linguistics*, page 299–311. Taylor and Francis Group Ltd., Oxford, 2017a.

Mark A Hasegawa-Johnson, Preethi Jyothi, Daniel McCloy, Majid Mirbagheri, Giovanni M di Liberto, Amit Das, Bradley Ekin, Chunxi Liu, Vimal Manohar, Hao Tang, et al. Asr for under-resourced languages from probabilistic transcription. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25 (1):50–63, 2017b.

Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL `http://dl.acm.org/citation.cfm?id=2132960.2132986`.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*, December 2017. URL `https://arxiv.org/abs/1712.05690`.

Petr Homola. A machine translation toolchain for polysynthetic languages. In *Proceedings of the 16th EAMT Conference*, 2012. URL `http://www.mt-archive.info/EAMT-2012-Homola.pdf`.

Steven A Jacobson. *Yup'ik Eskimo Dictionary*. Alaska Native Language Center, 1984.

Steven A. Jacobson. *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language, Preliminary Edition*. Alaska Native Language Center, Fairbanks, Alaska, 2nd edition, 2001.

Frederick Jelineck and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands, May 1980.

Robbie Jimerson, Kruthika Simha, Raymond Ptucha, and Emily Prudhommeaux. Improving ASR Output for Endangered Language Documentation. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, pages 187–191, 2018. doi: 10.21437/SLTU.2018-39. URL http://dx.doi.org/10.21437/SLTU.2018-39.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi kiu Lo, Darlene Stewart, and Jeffrey Micher. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of LREC-2020*, Marseille, France, May 2020.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://doi.org/10.1162/tacl_a_00065.

Lauri Karttunen. Finite-state constraints. In John Goldsmith, editor, *The Last Phonological Rule: Reflections on constraints and derivations*. University of Chicago Press, 1993.

Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Tranactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, March 1987.

Kimmo Kettunen. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245, 2014. doi: 10.1080/09296174.2014.911506. URL https://doi.org/10.1080/09296174.2014.911506.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/content/114/13/3521.

Judith Klavans, John Morgan, Stephen LaRocca, Jeffrey Micher, and Clare Voss. Challenges in speech recognition and translation of high-value low-density polysynthetic languages. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Papers)*, pages 283–293, Boston, MA, March 2018a. Association for Machine Translation in the Americas. URL https://www.aclweb.org/anthology/W18-1921.

Judith L Klavans, John Morgan, Stephen LaRocca, Jeffrey Micher, and Clare Voss. Challenges in speech recognition and translation of high-value low-density polysynthetic languages. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Papers)*, pages 283–293, 2018b.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P17-4012.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184, 1995.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P07-2045`.

Kimmo Koskenniemi. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki, 1983. URL `http://www.ling.helsinki.fi/koskenni/doc/Two-LevelMorphology.pdf`.

Anastasia Kuznetsova and Francis M. Tyers. A finite-state morphological analyser for paraguayan guaraní. *In submission*, 2019.

Jindřich Libovický and Jindřich Helcl. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2031. URL `https://www.aclweb.org/anthology/P17-2031`.

Patrick Littell, Chi-kiu Lo, Samuel Larkin, and Darlene Stewart. Multi-source transformer for Kazakh-Russian-English neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 267–274, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5326. URL `https://www.aclweb.org/anthology/W19-5326`.

Chi-kiu Lo. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL `https://www.aclweb.org/anthology/W19-5358`.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 682–701, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W18-6451`.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy, August 2019. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W19-5302`.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. Challenges of language technologies for the indigenous languages of the Americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1006`.

Manuel Mager, Elisabeth Mager, Alfonso Medina-Urrea, Ivan Vladimir Meza Ruiz, and Katharina Kann. Lost in translation: Analysis of information loss during machine translation between polysynthetic and fusional languages. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 73–83, Santa Fe, New Mexico, USA, August 2018b. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-4808`.

Joel Martin, Howard Johnson, Benoit Farley, and Anna Maclachlan. Aligning and using an English-Inuktitut parallel corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: Data driven machine translation and beyond, Volume 3*, pages 115–118. Association for Computational Linguistics, 2003.

Joel Martin, Rada Mihalcea, and Ted Pedersen. Word alignment for languages with scarce resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 65–74. Association for Computational Linguistics, 2005.

Thomas Mayer and Michael Cysouw. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/220_Paper.pdf`.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*, 2017.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. An Analysis of Neural Language Modeling at Multiple Scales. *arXiv preprint arXiv:1803.08240*, 2018.

Jeffrey Micher. Improving coverage of an Inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/W17-0114`.

Jeffrey Micher. Provenance and processing of an Inuktitut-English parallel corpus part 1: Inuktitut data preparation and factored data format. Technical Report ARL-TN-0923, US Army Research Laboratory, October 2018a. URL `https://www.arl.army.mil/arlreports/2018/technical-report.cfm?id=6182`.

Jeffrey Micher. Using the Nunavut Hansard data for experiments in morphological analysis and machine translation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 65–72, Santa Fe, New Mexico, USA, August 2018b. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-4807.pdf`.

Sabrina J. Mielke. Can you compare perplexity across different segmentations?, 2019. URL `https://sjmielke.com/comparing-perplexities.htm`.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. What kind of language is hard to language-model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1491. URL `https://www.aclweb.org/anthology/P19-1491`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL `http://dl.acm.org/citation.cfm?id=2999792.2999959`.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1045–1048, Makuhari, Chiba, Japan, September 2010. URL `https://www.isca-speech.org/archive/archive_papers/interspeech_2010/i10_1045.pdf`.

Mehryar Mohri. Language processing with weighted transducers. In *Proceedings of the 8th annual conference Traitement Automatique des Langues Naturelles (TALN 2001)*, 2001.

Christian Monson, Ariadna Font Llitjós, Roberto Aranovich, Lori Levin, Ralf Brown, Eric Peterson, Jaime Carbonell, and Alon Lavie. Building nlp systems for two resource-scarce indigenous languages: Mapudungun and quechua. *Strategies for developing machine translation for minority languages; 5th SALTMIL Workshop on Minority Languages*, page 15, 2006.

Kayo Nagai. *Mrs. Della Waghiyi's St. Lawrence Island Yupik Texts with Grammatical Analysis*. Number A2-006 in Endangered Languages of the Pacific Rim. Nakanishi Printing, Kyoto, Japan, 2001.

Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1103. URL `https://www.aclweb.org/anthology/D18-1103`.

Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependencies in stochastic language modeling. *Computer Speech and Language*, 8:1–38, 1994.

Yuta Nishimura, Katsuhito Sudoh, Graham Neubig, and Satoshi Nakamura. Multi-source neural machine translation with data augmentation. *CoRR*, abs/1810.06826, 2018. URL `http://arxiv.org/abs/1810.06826`.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Chris Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of Language Resources and Evaluation Conference (LREC'16)*, 2016.

Kemal Oflazer and Ilknur Durgar El-Kahlout. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W07/W07-0204`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://www.aclweb.org/anthology/P02-1040`.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://www.aclweb.org/anthology/N18-1202`.

Maja Popović. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL `https://www.aclweb.org/anthology/W15-3049`.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-6319`.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Mótliček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. The Kaldi speech recognition toolkit. In *Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011.

Matīss Rikters, Mārcis Pinnis, and Rihards Krišlauks. Training and adapting multilingual NMT for less-resourced and morphologically rich languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1595`.

Alexander Rudnick. *Cross-Lingual Word Sense Disambiguation for Low-Resource Hybrid Machine Translation*. PhD thesis, Indiana University, 12 2018. URL `https://scholarworks.iu.edu/dspace/handle/2022/22672`.

Lane Schwartz, Emily Chen, Benjamin Hunt, and Sylvia L.R. Schreiner. Bootstrapping a neural morphological analyzer for St. Lawrence Island Yupik from a finite-state transducer. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 87–96, Honolulu, February 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W19-6012`.

Lane Schwartz, Sylvia Schreiner, and Emily Chen. Community-focused language documentation in support of language education and revitalization for St. Lawrence Island Yupik. *Études Inuit Studies*, 2020. In press.

Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1021. URL `https://www.aclweb.org/anthology/P19-1021`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

Claude Shannon. Prediction and entropy of printed English. *Bell System Technical Journal*, 30:50–64, 1951. URL `http://www.princeton.edu/~wbialek/rome/refs/shannon_51.pdf`.

Miikka Silfverberg and Francis Tyers. Data-driven morphological analysis for Uralic languages. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 1–14, 2019.

Grace Slwooko. *Sivuqam Ungipaghaatangi I*. University of Alaska, Anchorage, AK, 1977.

Grace Slwooko. *Sivuqam Ungipaghaatangi II*. University of Alaska, Anchorage, AK, 1979.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden, 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-2006.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216, January 1990. URL `https://doi.org/10.1016/0004-3702(90)90007-M`.

Statistics Canada. Census in Brief: The Aboriginal languages of First Nations people, Métis and Inuit, 2017. URL `https://www12.statcan.gc.ca/census-recensement/2016/as-sa/98-200-x/2016022/98-200-x2016022-eng.cfm`.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf`.

The Crow Language Conservancy. Crow dictionary online, 2019. URL `https://dictionary.crowlanguage.org`.

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

J. N. Washington, I. Salimzyanov, and F. M. Tyers. Finite-state morphological transducers for three kypchak languages. In *Proceedings of the 9th Conference on Language Resources and Evaluation, LREC2014*, 2014.

Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probability of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, July 1991.

Wycliffe. *Yupik New Testament*. Wycliffe Bible Translators, Saint Lawrence Island, Alaska, 2018.

Barret Zoph and Kevin Knight. Multi-source neural translation. In *Proceedings of NAACL-HLT*, pages 30–34, 2016.