Topics in Sequence-to-Sequence Learning for Natural Language Processing

Roee Aharoni

Ph.D. Thesis

Submitted to the Senate of Bar-Ilan University

Ramat Gan, Israel May 2020

This work was carried out under the supervision of Prof. Yoav Goldberg, Department of Computer Science, Bar-Ilan University.

To my parents, Shoshana (Shoshi) and Yehoshua (Shuki). Thank you for all the endless love, support, and for raising me to always be curious. To Sapir, my best friend and partner in life. Thank you for coping with all the busy weekends, long travels and white nights before deadlines, and for all your love and support during this journey.

This work is dedicated to you.

Acknowledgements

To the one and only Yoav Goldberg, my advisor in the journey that is concluded in this work. Thank you for all the patience, inspiration, endless knowledge, creativity, and general wizardry and super powers. You made this journey a true pleasure, and those several years one of the most enjoyable and rewarding periods of my life. Other than on NLP, I learned so much from you on how to do proper science, how to ask the right questions, and how to communicate ideas clearly and concisely. I couldn't ask for a better advisor and truly cherish all the invaluable things I learned from you. Thank you!

To all BIU-NLP members, and particularly Ido Dagan, Eliyahu Kiperwasser, Vered Schwartz, Gabriel Stanovsky, Yanai Elazar and Amit Moryessef – thank you for being amazing colleagues and for creating a supporting and welcoming environment, which makes research a truly amazing experience. I learned so much from you and I'm sure we will collaborate again in the future.

To Orhan Firat, Melvin Johnson and the rest of the Google Translate team – thank you for sharing the opportunity to work with you in one of the most impactful natural language processing teams in the world. Summer 2018 was an unforgettable one and really a dream coming true, which certainly shaped my how my future will look like in a significant way.

Last but not least, I would like to thank all the teachers that introduced and helped me deepen my practice of Yoga. This practice had a significant contribution to keeping me strong and flexible, also physically but mostly mentally, during this journey. Namaste.

Preface

Publications

Portions of this thesis are joint work and have been published elsewhere.

Chapter 3, "Improving Sequence-to-Sequence Learning for Morphological Inflection Generation" appeared in the proceedings of the 14th Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON 2016), Berlin, Germany (Aharoni et al., 2016).

Chapter 3, "Morphological Inflection Generation with Hard Monotonic Attention" appeared in the proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, Canada (Aharoni and Goldberg, 2017a).

Chapter 4, "Towards String-to-Tree Neural Machine Translation", appeared in the proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, Canada (Aharoni and Goldberg, 2017b).

Chapter 5, "Split and Rephrase: Better Evaluation and a Stronger Baseline", appeared in the proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Melbourne, Australia (Aharoni and Goldberg, 2018).

Chapter 6, "Massively Multilingual Neural Machine Translation", appeared in the proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

(NAACL-HLT 2019), Minneapolis, USA (Aharoni et al., 2019).

Chapter 7, "Emerging Domain Clusters in Pretrained Language Models", appeared in the proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Seattle, USA (Aharoni and Goldberg, 2020).

Funding

This work was supported by Intel (via the ICRI-CI grant), the Israeli Science Foundation (grant number 1555/15), the German Research Foundation via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1) and by Google.

Contents

Abstract

Making computers successfully process natural language is a long standing goal in artificial intelligence that spreads across numerous tasks and applications. One prominent example is machine translation, that has preoccupied the minds of scientists for many decades (Bar-Hillel (1951); Weaver (1955)). While serving as a scientific benchmark for the progress of artificial intelligence, machine translation and many other natural language processing (NLP) applications are also highly useful for millions of people around the globe, enabling better communication and easier access to the world's information.

Many NLP tasks can be cast as sequence-to-sequence learning problems – e.g. problems that involve sequential input and output. From a machine learning perspective, such problems handle the prediction of a structured output given a structured input, as each element of the output sequence is usually predicted while conditioning on the input sequence and on the previously predicted elements. This setting requires rich feature representations and specialized inference algorithms to take into account the different interactions between and within the input and output elements.

The recent proliferation of neural-network based machine learning methods (also known as "deep-learning") has enabled profound progress on sequence-to-sequence learning tasks; specifically, it allowed to implicitly learn representations in an end-to-end manner, without requiring manual feature engineering as was common in previous methods. In addition, the limitation of using a fixed context window when modeling each element due to computational burden was removed with the neural methods, allowing for much better modeling of long-range dependencies in such tasks.

In order to unlock the full potential of neural-network based methods for NLP applications, many new research questions arise – can we design neural models while integrating existing knowledge about language? Can we take advantage of such implicitly learned representations in shared multilingual settings? What are the limitations of such models? What can we learn about textual domains from the representations such models learn?

In this thesis, I seek answers to those questions revolving around neural sequence-to-sequence learning for NLP. My works involve different levels of language studies: Morphology, the study of words, how they are formed, and their relationship to other words in the same language, where I propose novel neural architectures for inflection generation; Syntax – the set of rules, principles, and processes that govern the structure of sentences in a given language, where I study the integration of syntactic information to neural machine translation; Semantics – the study of meaning in language, usually at the sentence level, where I worked on complex sentence simplification that preserves the input semantics, and on massively multilingual translation that encodes dozens of languages to a shared semantic space; and finally Pragmatics – that looks at linguistic context beyond the sentence level, where I proposed a novel method to select training data using contextualized sentence representations from pretrained neural language models.

In Chapter 3, "Morphological Inflection Generation with Hard Monotonic Attention", I propose novel neural architectures for sequence-to-sequence learning that explicitly model a monotonic alignment between the source and target sequence elements. The models are inspired by the monotonic alignment between the characters in different morphological inflections of a given word.

I evaluate the proposed models across multiple datasets for morphological inflection generation in various languages, where they obtain state-of-the-art results. My proposed approach became a well-known baseline in the literature, and is still considered as a state-of-the-art approach for such morphological tasks.

In Chapter 4, "Towards String-to-Tree Neural Machine Translation", I propose a method to incorporate linguistic knowledge into neural sequence-to-

sequence learning models for machine translation. Inspired by works on syntactic parsing, I suggest to represent the target sentence as a lexicalized, linearized constituency parse tree.

I show that incorporating such knowledge improves the translation quality as measured in BLEU and by human raters, especially in low-resource scenarios, across multiple language pairs. This work is one of the first attempts to incorporate such linguistic knowledge into end-to-end neural models, which started an ongoing line of work in the MT and NLP community. Subsequent works explored other types of syntactic annotations, or other ways to inject linguistic knowledge.

In Chapter 5, "Split and Rephrase: Better Evaluation and a Stronger Baseline", I investigate the ability of neural sequence-to-sequence learning models to perform semantic text simplification where the input is a long, complex sentence and the output is several shorter sentences that convey the same meaning.

I show that while such models seem to provide state-of-the-art results on the proposed benchmark, they are prone to over-fitting and memorization. I then propose a new data split based on the structured semantic relations that describe the sentences, to better test the generalization abilities of such models – unveiling their limitations. This work was later extended to a larger, more realistic dataset by others, which adopted our proposed approaches.

In Chapter 6, "Massively Multilingual Neural Machine Translation", I investigate scaling neural machine translation (NMT) models to massively multilingual settings, involving up to 103 languages and more than 95 million sentence pairs in a single model.

I show that training such models is highly effective for improving the performance on low-resource language pairs, resulting in state-of-the-art results on the publicly available TED talks dataset. I then conduct large-scale experiments where I point at the trade-off between the degradation in supervised translation quality due to the bottleneck caused by scaling to numerous languages vs. improved generalization abilities in zero-shot translation when increasing the number of languages. While this work was the first to scale NMT mod-

els to such settings, many subsequent works now train massively multilingual language models that enable cross-lingual transfer learning, for better NLP in under-resourced languages.

In Chapter 7, "Emerging Domain Clusters in Pretrained Language Models", I investigate sentence representations learned by various large scale neural language models with respect to the *domains* those sentences were drawn from. I show that using such representations, it is possible to cluster sentences into domains with very high accuracy, in a purely unsupervised manner.

I then propose ways to utilize these emerging domain clusters to select data for training domain-specific neural machine translation models. I show that such models outperform strong baselines that are either trained on all the available data or use established data selection methods. This work is the first to explore such use of pretrained language models for sentence clustering or domain data selection, and suggests a novel, pragmatic, data-driven approach to the notion of domains in textual data.

As this work shows, neural sequence-to-sequence learning is a powerful approach to tackle many different problems in various areas of NLP – from the low-level morphological tasks to the high-level task of translating between numerous languages. While being highly effective, there are still many areas left to explore further. In the final part of this thesis, I conclude with the contributions of this work and with a list of directions for future work which I find important to pursue further on the subject.

Chapter 1

Introduction

In this chapter we provide an introduction to the different chapters in this thesis, including our contributions and related work.

1.1 Hard Attention Architectures for Morphological Inflection Generation

Morphological inflection generation involves generating a target word (e.g. "härtestem", the German word for "hardest"), given a source word (e.g. "hart", the German word for "hard") and the morpho-syntactic attributes of the target (POS=adjective, gender=masculine, type=superlative, etc.).

The task is important for many down-stream NLP tasks such as machine translation, especially for dealing with data sparsity in morphologically rich languages where a lemma can be inflected into many different word forms. Several studies have shown that translating into lemmas in the target language and then applying inflection generation as a post-processing step is beneficial for phrase-based machine translation (Minkov et al., 2007; Toutanova et al., 2008; Clifton and Sarkar, 2011; Fraser et al., 2012; Chahuneau et al., 2013) and more recently for neural machine translation (García-Martínez et al., 2016).

The task was traditionally tackled with hand engineered finite state trans-

ducers (FST) (Koskenniemi, 1984; Kaplan and Kay, 1994) which rely on expert knowledge, or using trainable weighted finite state transducers (Mohri et al., 1997; Eisner, 2002) which combine expert knowledge with data-driven parameter tuning. Many other machine-learning based methods (Yarowsky and Wicentowski, 2000; Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Hulden et al., 2014; Ahlberg et al., 2015; Nicolai et al., 2015) were proposed for the task, although with specific assumptions about the set of possible processes that are needed to create the output sequence.

More recently, the task was modeled as neural sequence-to-sequence learning over character sequences with impressive results (Faruqui et al., 2016). The vanilla encoder-decoder models as used by Faruqui et al. compress the input sequence to a single, fixed-sized continuous representation. Instead, the soft-attention based sequence to sequence learning paradigm (Bahdanau et al., 2015) allows directly conditioning on the entire input sequence representation, and was utilized for morphological inflection generation with great success (Kann and Schütze, 2016a,b).

However, the neural sequence-to-sequence models require large training sets in order to perform well: their performance on the relatively small CELEX dataset is inferior to the latent variable WFST model of Dreyer et al. (2008). Interestingly, the neural WFST model by Rastogi et al. (2016) also suffered from the same issue on the CELEX dataset, and surpassed the latent variable model only when given twice as much data to train on.

In Chapter 3, we propose a model which handles the above issues by directly modeling an almost monotonic alignment between the input and output character sequences, which is commonly found in the morphological inflection generation task (e.g. in languages with concatenative morphology). The model consists of an encoder-decoder neural network with a dedicated control mechanism: in each step, the model attends to a *single* input state and either writes a symbol to the output sequence or advances the attention pointer to the next state from the bi-directionally encoded sequence.

This modeling suits the natural monotonic alignment between the input and output, as the network learns to attend to the relevant inputs before writing the

output which they are aligned to. The encoder is a bi-directional RNN, where each character in the input word is represented using a concatenation of a forward RNN and a backward RNN states over the word's characters. The combination of the bi-directional encoder and the controllable hard attention mechanism enables to condition the output on the entire input sequence. Moreover, since each character representation is aware of the neighboring characters, non-monotone relations are also captured, which is important in cases where segments in the output word are a result of long range dependencies in the input word. The recurrent nature of the decoder, together with a dedicated feedback connection that passes the last prediction to the next decoder step explicitly, enables the model to also condition the current output on all the previous outputs at each prediction step.

The hard attention mechanism allows the network to jointly align and transduce while using a focused representation at each step, rather then the weighted sum of representations used in the soft attention model. This makes our model *resolution preserving* (Kalchbrenner et al., 2016) while also keeping decoding time linear in the output sequence length rather than multiplicative in the input and output lengths as in the soft-attention model. In contrast to previous sequence-to-sequence work, we do not require the training procedure to also learn the alignment. Instead, we use a simple training procedure which relies on independently learned character-level alignments, from which we derive gold transduction+control sequences. The network can then be trained using straightforward cross-entropy loss.

To evaluate our model, we perform extensive experiments on three previously studied morphological inflection generation datasets: the CELEX dataset (Baayen et al., 1993), the Wiktionary dataset (Durrett and DeNero, 2013) and the SIGMORPHON2016 dataset (Cotterell et al., 2016). We show that while our model is on par with or better than the previous neural and non-neural state-of-the-art approaches, it also performs significantly better with very small training sets, being the first neural model to surpass the performance of the weighted FST model with latent variables which was specifically tailored for the task by Dreyer et al. (2008). Finally, we analyze and compare our model and the soft

attention model, showing how they function very similarly with respect to the alignments and representations they learn, in spite of our model being much simpler. This analysis also sheds light on the representations such models learn for the morphological inflection generation task, showing how they encode specific features like a symbol's type and the symbol's location in a sequence.

To summarize, our contributions in Chapter 3 are three-fold:

1. We present a hard attention model for nearly-monotonic sequence to sequence learning, as common in the morphological inflection setting.
2. We evaluate the model on the task of morphological inflection generation, establishing a new state of the art on three previously-studied datasets for the task.
3. We perform an analysis and comparison of our model and the soft-attention model, shedding light on the features such models extract for the inflection generation task.

Our Hard-Attention models were adopted by the community as a strong tool for morphological inflection generation (Gorman et al., 2019), lemmatization (Şahin and Gurevych, 2019), surface realization (Puzikov et al., 2019), and machine translation (Press and Smith, 2018), among others. They were also extended to non-monotonic scenarios (Wu et al., 2018a) and to exact inference (Wu and Cotterell, 2019). Our models were improved further resulting in the winning submissions for the 2017 CoNLL shared task on morphological reinflection (Makarov et al., 2017; Cotterell et al., 2017) and the 2018 CoNLL shared task on Universal Morphological Reinflection (Makarov and Clematide, 2018; Cotterell et al., 2018).

1.2 Linearizing Syntax for String-to-Tree Neural Machine Translation

Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014a; Bahdanau et al., 2015) has recently became the state-of-the-art ap-

proach to machine translation (Bojar et al., 2016), while being much simpler than the previously dominant phrase-based statistical machine translation (SMT) approaches (Koehn, 2010). NMT models usually do not make explicit use of syntactic information about the languages at hand.

However, a large body of work was dedicated to syntax-based SMT (Williams et al., 2016). One prominent approach to syntax-based SMT is string-to-tree (S2T) translation (Yamada and Knight, 2001, 2002), in which a source-language string is translated into a target-language tree. S2T approaches to SMT help to ensure the resulting translations have valid syntactic structure, while also mediating flexible reordering between the source and target languages. The main formalism driving current S2T SMT systems is GHKM rules (Galley et al., 2004, 2006), which are synchronous transduction grammar (STSG) fragments, extracted from word-aligned sentence pairs with syntactic trees on one side. The GHKM translation rules allow flexible reordering on all levels of the parse-tree.

In Chapter 4, we suggest that NMT can also benefit from the incorporation of syntactic knowledge, and propose a simple method of performing string-to-tree neural machine translation. Our method is inspired by recent works in syntactic parsing, which model trees as sequences (Vinyals et al., 2015; Choe and Charniak, 2016). Namely, we translate a source sentence into a linearized, lexicalized constituency tree.

The linearized trees we predict are different in their structure from those in Vinyals et al. (2015) as instead of having part of speech tags as terminals, they contain the words of the translated sentence. We intentionally omit the POS information as including it would result in significantly longer sequences. The S2T model is trained on parallel corpora in which the target sentences are automatically parsed. Since this modeling keeps the form of a sequence-to-sequence learning task, we can employ the conventional attention-based sequence to sequence paradigm Bahdanau et al. (2015) as-is, while enriching the output with syntactic information.

Some recent works did propose to incorporate syntactic or other linguistic knowledge into NMT systems, although mainly on the source side: Eriguchi

et al. (2016a,b) replace the encoder in an attention-based model with a Tree-LSTM (Tai et al., 2015) over a constituency parse tree; Bastings et al. (2017) encoded sentences using graph-convolutional networks over dependency trees; Sennrich and Haddow (2016) proposed a factored NMT approach, where each source word embedding is concatenated to embeddings of linguistic features of the word; Luong et al. (2015a) incorporated syntactic knowledge via multi-task sequence to sequence learning: their system included a single encoder with multiple decoders, one of which attempts to predict the parse-tree of the source sentence; Stahlberg et al. (2016) proposed a hybrid approach in which translations are scored by combining scores from an NMT system with scores from a Hiero (Chiang, 2005, 2007) system. Shi et al. (2016) explored the syntactic knowledge encoded by an NMT encoder, showing the encoded vector can be used to predict syntactic information like constituency trees, voice and tense with high accuracy.

In parallel and highly related to our work, Eriguchi et al. (2017) proposed to model the target syntax in NMT in the form of dependency trees by using an RNNG-based decoder Dyer et al. (2016), while Nădejde et al. (2017) incorporated target syntax by predicting CCG tags serialized into the target translation. Our work differs from those by modeling syntax using constituency trees, as was previously common in the "traditional" syntax-based machine translation literature.

We show our method improves the performance as measured by BLEU in both high and low-resource settings, across three language pairs. We also perform a human evaluation in the high resource scenario where we show human rates prefer the outputs of the string-to-tree system over a similarly trained string-to-string system.

To summarize, our contributions in Chapter 4 are as follows:

- We propose a method for performing string-to-tree neural machine translation by linearizing constituency syntax trees.
- We show our method improves the performance as measured by BLEU in both high and low-resource settings, across three language pairs, and also

generates better translations according to human raters.

- We perform an extensive analysis of the outputs generated by our method, showing that they produce valid syntactic trees, include more reordering, and generate more relative pronouns.

This work and others initiated a line of work on representing syntactic trees and trees in general using neural sequence to sequence models. Some examples include dependency-based NMT (Wu et al., 2018b), syntactically supervised transformers for faster NMT (Akoury et al., 2019), and Forest-based NMT Ma et al. (2018). Regarding tasks other than MT, examples include translating between different semantic formalisms (Stanovsky and Dagan, 2018), code generation (Alon et al., 2019) and response generation Du and Black (2019).

1.3 Semantic Sentence Simplification by Splitting and Rephrasing

Processing long, complex sentences is challenging. This is true either for humans in various circumstances (Inui et al., 2003; Watanabe et al., 2009; De Belder and Moens, 2010) or in NLP tasks like parsing (Tomita, 1986; McDonald and Nivre, 2011; Jelínek, 2014) and machine translation (Chandrasekar et al., 1996; Pouget-Abadie et al., 2014; Koehn and Knowles, 2017). An automatic system capable of breaking a complex sentence into several simple sentences that convey the same meaning is very appealing.

Narayan et al. (2017) introduced a dataset, evaluation method and baseline systems for the task, naming it "Split and Rephrase". The dataset includes 1,066,115 instances mapping a single complex sentence to a sequence of sentences that express the same meaning, together with RDF triples that describe their semantics.

They considered two system setups: a text-to-text setup that does not use the accompanying RDF information, and a semantics-augmented setup that does. We focus on the text-to-text setup, which we find to be more challenging and

more natural.

We begin with vanilla sequence-to-sequence models with attention (Bahdanau et al., 2015) and reach an accuracy of 77.5 BLEU, substantially outperforming the text-to-text baseline of Narayan et al. (2017) and approaching their best RDF-aware method. However, manual inspection reveal many cases of unwanted behaviors in the resulting outputs: (1) many resulting sentences are *unsupported* by the input: they contain correct facts about relevant entities, but these facts were not mentioned in the input sentence; (2) some facts are *repeated*—the same fact is mentioned in multiple output sentences; and (3) some facts are *missing*—mentioned in the input but omitted in the output.

The model learned to *memorize entity-fact pairs* instead of learning to split and rephrase. Indeed, feeding the model with examples containing entities alone without any facts about them causes it to output perfectly phrased but unsupported facts. Digging further, we find that 99% of the simple sentences (more than 89% of the unique ones) in the validation and test sets also appear in the training set, which—coupled with the good memorization capabilities of sequence-to-sequence models and the relatively small number of distinct simple sentences—helps to explain the high BLEU score.

To aid further research on the task, we propose a more challenging split of the data. We also establish a stronger baseline by extending the sequence-to-sequence approach with a copy mechanism, which was shown to be helpful in similar tasks Gu et al. (2016); Merity et al. (2017); See et al. (2017). On the original split, our models outperform the best baseline of Narayan et al. (2017) by up to 8.68 BLEU, without using the RDF triples. On the new split, the vanilla SEQ2SEQ models break completely, while the copy-augmented models perform better. In parallel to our work, an updated version of the dataset was released (v1.0), which is larger and features a train/test split protocol which is similar to our proposal. We report results on this dataset as well.

To summarize, our contributions in Chapter 5 are as follows:

- We point at issues in the training data for the task, and show how they make it easy for simple neural sequence-to-sequence models to perform

well by memorizing the training set.

- We create a new data split for the task by taking into account the underlying semantic relations, which makes it much harder to solve it by memorization.
- We establish stronger baselines by using models with a copy mechanism, allowing better generalization.

A consequent work by Botha et al. (2018) created a larger, more natural dataset for the task. Using our proposed modeling recipes, they greatly improved the performance on the task using their new dataset, encouraging future work on the task.

1.4 Massively Multilingual Neural Machine Translation: Towards Universal Translation

Neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2015; Sutskever et al., 2014b) is the current state-of-the-art approach for machine translation in both academia (Bojar et al., 2016, 2017, 2018) and industry (Wu et al., 2016; Hassan et al., 2018). Recent works (Dong et al., 2015; Firat et al., 2016a; Ha et al., 2016; Johnson et al., 2017) extended the approach to support multilingual translation, i.e. training a single model that is capable of translating between multiple language pairs.

Multilingual models are appealing for several reasons. First, they are more efficient in terms of the number of required models and model parameters, enabling simpler deployment. Another benefit is transfer learning; when low-resource language pairs are trained together with high-resource ones, the translation quality may improve (Zoph et al., 2016; Nguyen and Chiang, 2017). An extreme case of such transfer learning is zero-shot translation (Johnson et al., 2017), where multilingual models are able to translate between language pairs that were never seen during training.

While very promising, it is still unclear how far one can scale multilingual NMT in terms of the number of languages involved. Previous works on multilingual NMT typically trained models with up to 7 languages (Dong et al., 2015; Firat et al., 2016b; Ha et al., 2016; Johnson et al., 2017; Gu et al., 2018a) and up to 20 trained directions (Cettolo et al., 2017) simultaneously. One recent exception is Neubig and Hu (2018) who trained many-to-one models from 58 languages into English. While utilizing significantly more languages than previous works, their experiments were restricted to many-to-one models in a low-resource setting with up to 214k examples per language-pair and were evaluated only on four translation directions.

In Chapter 6, we take a step towards practical "universal" NMT – training massively multilingual models which support up to 102 languages and with up to one million examples per language-pair simultaneously. Specifically, we focus on training "English-centric" many-to-many models, in which the training data is composed of many language pairs that contain English either on the source side or the target side. This is a realistic setting since English parallel data is widely available for many language pairs. We restrict our experiments to Transformer models (Vaswani et al., 2017) as they were shown to be very effective in recent benchmarks (Ott et al., 2018), also in the context of multilingual models (Lakew et al., 2018; Sachan and Neubig, 2018).

We evaluate the performance of such massively multilingual models while varying factors like model capacity, the number of trained directions (tasks) and low-resource vs. high-resource settings. Our experiments on the publicly available TED talks dataset Qi et al. (2018) show that massively multilingual many-to-many models with up to 58 languages to-and-from English are very effective in low resource settings, allowing to use high-capacity models while avoiding overfitting and achieving superior results to the current state-of-the-art on this dataset Neubig and Hu (2018); Wang et al. (2018) when translating into English.

We then turn to experiment with models trained on 103 languages in a high-resource setting. For this purpose we compile an English-centric in-house dataset, including 102 languages aligned to-and-from English with up to one million

examples per language pair. We then train a single model on the resulting 204 translation directions and find that such models outperform strong bilingual baselines by more than 2 BLEU averaged across 10 diverse language pairs, both to-and-from English. Finally, we analyze the trade-offs between the number of involved languages and translation accuracy in such settings, showing that massively multilingual models generalize better to zero-shot scenarios. We hope these results will encourage future research on massively multilingual NMT.

In summary, our contributions in Chapter 6 are the following:

- We perform extensive experiments on massively multilingual NMT, showing that a single Transformer model can successfully scale to 103 languages and 204 trained directions with one million examples per direction, while outperforming strong single-pair baselines.
- We show that massively multilingual models are effective in low-resource settings, allowing to use high-capacity models while avoiding overfitting due to the data-scarcity in single-pair settings.
- We analyze the trade-offs between model capacity, the number of training examples and the number of languages involved in such settings.

While this work was the first to scale NMT models to such settings, many subsequent works now train massively multilingual language models that enable cross-lingual transfer learning, for better NLP in under-resourced languages (Devlin et al., 2019; Conneau et al., 2019; Siddhant et al., 2019). Other subsequent works investigate scaling such models even further in terms of the number of parameters and training examples (Arivazhagan et al., 2019) and analyzed the emerging language families within their learned representations with respect to linguistic theories on the subject (Kudugunta et al., 2019). Improving such models and making them available to the public is of very high importance and has global impact, as most of the current research on NLP is mainly focused on English (Bender, 2019).

1.5 Emerging Domain Clusters in Pretrained Language Models

In Chapter 7, we investigate the use of large pre-trained neural language models for the task of *domain data selection* for machine translation.

It is common knowledge in modern NLP that using large amounts of high-quality training data is a key aspect in building successful machine-learning based systems. For this reason, a major challenge when building such systems is obtaining data in the domain of interest. But what defines a domain? Natural language varies greatly across topics, styles, levels of formality, genres and many other linguistic nuances (van der Wees et al., 2015; van der Wees, 2017; Niu et al., 2017). This overwhelming diversity of language makes it hard to find the right data for the task, as it is nearly impossible to well-define the exact requirements from such data with respect to all the aforementioned aspects. On top of that, domain labels are usually unavailable – e.g. in useful large-scale web-crawled data like Common Crawl (Raffel et al., 2019).[1]

Domain data selection is the task of selecting the most appropriate data for a domain from a large corpus given a smaller set of in-domain data (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013; Silva et al., 2018). We propose to use the recent, highly successful self-supervised pre-trained language models, e.g. Devlin et al. (2019); Liu et al. (2019) for domain data selection. As pretrained LMs demonstrate state-of-the-art performance across many NLP tasks after being trained on massive amounts of data, we hypothesize that the robust representations they learn may be useful for mapping sentences to domains in an unsupervised, data-driven approach. We show that these models indeed learn to cluster sentence representations to domains without further supervision, and quantify this by fitting Gaussian Mixture Models (GMMs) to the learned representations and measuring the purity of the resulting clustering. We then propose methods to leverage these emergent domain clusters for domain data selection in two ways:

[1] https://commoncrawl.org/

- Via distance-based retrieval in the sentence embedding space induced by the pretrained language model.
- By fine-tuning the pretrained language model for binary classification, where positive examples are from the domain of interest.

Our methods enable to select relevant data for the task while requiring only a small set of monolingual in-domain data. As they are based solely on the representations learned by self-supervised LMs, they do not require additional domain labels which are usually vague and over-simplify the notion of domain in textual data.

We evaluate our method on data selection for neural machine translation (NMT) using the multi-domain German-English parallel corpus composed by Koehn and Knowles (2017). Our data selection methods enable to train NMT models that outperform those trained using the well-established cross-entropy difference method of Moore and Lewis (2010) across five diverse domains, achieving a recall of more than 95% in all cases with respect to an oracle that selects the "true" in-domain data.

To summarize, our contributions in Chapter 7 are as follows.

- We show that pre-trained language models are highly capable of clustering textual data to domains with high accuracy in a purely unsupervised manner.
- We propose methods to select in-domain data based on this property using vector-space retrieval and positive-unlabeled fine-tuning of pretrained language models for binary classification.
- We show the applicability of our proposed data selection methods on a popular benchmark for domain adaptation in machine translation.
- An additional contribution is a new, improved data split we create for this benchmark, as we point on issues with previous splits used in the literature.

We hope this work will encourage more research on understanding the data landscape in NLP, enabling to "find the right data for the task" in the age of massive models and diverse data sources.

1.6 Outline

The next chapter covers more background and related work on neural networks and neural sequence-to-sequence learning. Chapters 3 to 7 are the main body of the work and describe the aforementioned research works in detail. Finally, Chapter 8 describes conclusions and directions for future work on the subject.

Chapter 2

Background

We begin by laying some background on neural sequence-to-sequence learning, and presenting related work.

2.1 Neural Networks Basics

Notation We use bold, lower-case letters for vectors (e.g. $\boldsymbol{v}$) and bold, upper-case letters for matrices (e.g. $\boldsymbol{W}$).

As our basic building blocks, we will begin with describing two types of neural networks: Feed-Forward neural networks, a.k.a. Multi-Layer-Perceptrons (MLP's) and Recurrent Neural Networks (RNN's).

2.1.1 Feed-Forward Neural Networks

A feed-forward, single-layer neural network can be defined as a parameterized, non-linear function f mapping an input vector $\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{x}|}$ into an output vector $\boldsymbol{y} \in \mathbb{R}^{|\boldsymbol{y}|}$ by computing:

$$\boldsymbol{y} = f_1(\boldsymbol{x}) = z(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) \tag{2.1}$$

where z is an element-wise non-linear function (e.g. tanh or the sigmoid function, among others), and $\boldsymbol{W}_1 \in \mathbb{R}^{|\boldsymbol{y}|\times|\boldsymbol{x}|}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{|\boldsymbol{y}|}$ are the parameters of f. Multi-layer neural networks are compositions of functions of this form. For example, a two-layer feed-forward neural network may be of the form:

$$\boldsymbol{y} = f(\boldsymbol{x}) = f_2(f_1(\boldsymbol{x})) = z(\boldsymbol{W}_2(z(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2)) \tag{2.2}$$

2.1.2 Recurrent Neural Networks

A caveat of feed-forward networks is their inability to model an unbounded, variable-sized input, since their input is a fixed-sized vector. This is especially important in NLP, where we often need to model sentences which vary in length. A recurrent neural network maps a sequence of input vectors: $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_t$ into a sequence of output vectors: $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \ldots, \boldsymbol{h}_t$, allowing variable-length input. A common architecture for an RNN is the Elman RNN (Elman, 1990), which describes the following computation:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \tag{2.3}$$

where $\boldsymbol{W} \in \mathbb{R}^{D_{RNN}\times|\boldsymbol{x}_t|}, \boldsymbol{U} \in \mathbb{R}^{D_{RNN}\times D_{RNN}}$ and $\boldsymbol{b} \in \mathbb{R}^{D_{RNN}}$ are the RNN's parameters. In practice, the Elman RNN is hard to train due to the "vanishing gradient" issue (Bengio et al., 1994; Hochreiter, 1998; Pascanu et al., 2013). A popular RNN architecture which is less sensitive to this issue is the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), which is defined as follows:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{\boldsymbol{i}}[\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_{\boldsymbol{i}}) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_{\boldsymbol{o}}[\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_{\boldsymbol{o}}) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_{\boldsymbol{f}}[\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_{\boldsymbol{f}}) \\
\tilde{\boldsymbol{c}}_t &= \tanh\left(\boldsymbol{W}_{\tilde{\boldsymbol{c}}}[\boldsymbol{x}_t, \boldsymbol{h}_{t-1}] + \boldsymbol{b}_{\tilde{\boldsymbol{c}}}\right) \\
\boldsymbol{c}_t &= \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{c}_t\right)
\end{aligned} \tag{2.4}$$

Another widely adopted RNN architecture is the Gated Recurrent Unit (GRU) (Cho et al., 2014; Chung et al., 2014) which is slightly simpler than the LSTM:

$$\begin{aligned} \boldsymbol{z}_t &= \sigma(\boldsymbol{W}_z[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]) \\ \boldsymbol{r}_t &= \sigma(\boldsymbol{W}_r[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]) \\ \tilde{\boldsymbol{h}}_t &= \tanh\left(\boldsymbol{W}[\boldsymbol{r}_t \circ \boldsymbol{h}_{t-1}, \boldsymbol{x}_t]\right) \\ \boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \circ \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \circ \tilde{\boldsymbol{h}}_t \end{aligned} \tag{2.5}$$

2.2 Neural Machine Translation

Neural Machine Translation (NMT) is an approach for machine translation, which models the task using neural networks as the learning algorithm. In this section we will formally define the task and present some of the basic components in a common NMT system.

2.2.1 Task Definition

We begin by formally defining the machine translation task. Given a source sentence: $F = f_1, f_2, f_3, \ldots, f_{|F|}$, we would like to find a translation $E = e_1, e_2, e_3, \ldots, e_{|E|}$. Our task is to find a function mt for which: $\text{mt}(F) = \hat{E}$, where $\hat{E}$ is a translation of F.

In Statistical Machine Translation (SMT), we would like to create a probabilistic model which estimates $p(E|F, \theta)$ where θ is a set of parameters learned from example translations (parallel corpora). With such a model in hand one can find a translation by searching for $\text{mt}(F) = \arg\max_E p(E|F, \theta)$.

Neural Machine Translation In neural machine translation, we would like to model $p(E|F, \theta)$ using a neural network. We will now describe the structure of such network and how it is trained.

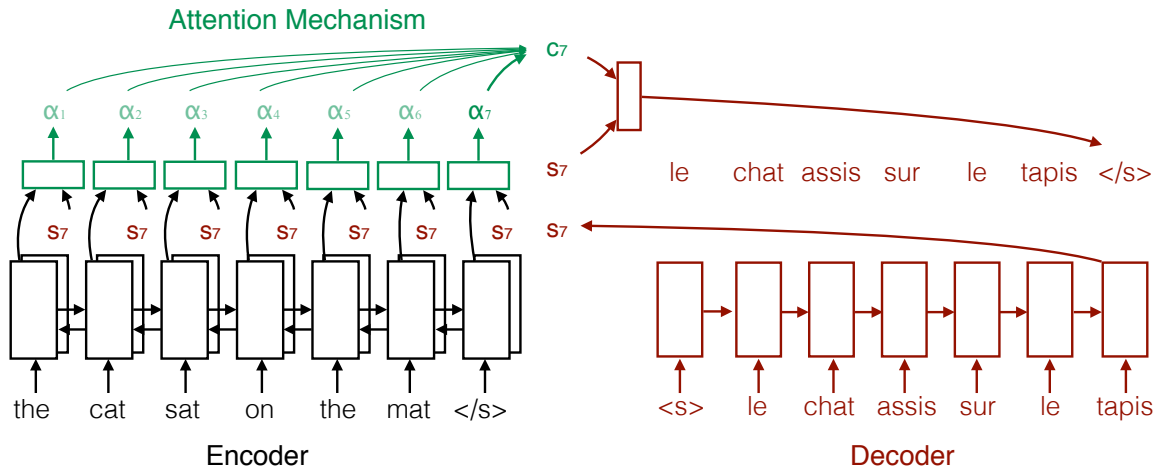


Figure 2.1: An illustration of an NMT model with a bidirectional RNN encoder, an RNN decoder and an attention mechanism.

2.2.2 Dense Word Representations

In order to perform the translation we first need to define a representation for the words, or tokens, in the input and output sentences. These representations will be fed into the neural network. Each source word $w_i \in \Sigma_{input}$ where Σ_{input} is the input vocabulary of size $|\Sigma_{input}|$ will be represented using a fixed-sized vector $\boldsymbol{w}_i$ of size D_{input}, and similarly for the output words: for each word v_j in the output vocabulary Σ_{output} we will have a vector $\boldsymbol{v}_i$, where $|\boldsymbol{v}_i| = D_{output}$. Note that this definition results in two matrices: $\boldsymbol{W} \in \mathbb{R}^{|\Sigma_{input}| \times D_{input}}$ which we will call the input embedding matrix, and $\boldsymbol{U} \in \mathbb{R}^{|\Sigma_{output}| \times D_{output}}$ which we will call the output embedding matrix.

2.2.3 Encoder

Once we defined the dense word representations which represent the words in the vocabularies, we will now define the first part of the network, which we name the Encoder. The Encoder's role is to create a representation for the entire input sentence, which will capture the interactions and dependencies between the different words in it. In general, we can define the Encoder as a parameterized function that receives a sequence of input embeddings: $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \ldots, \boldsymbol{f}_N$

and outputs a sequence of annotation vectors: $\hat{f} = \hat{f}_1, \hat{f}_2, \hat{f}_3, \ldots, \hat{f}_N$, where each annotation vector is of a fixed size $D_{encoder}$. While there are many different encoder variations in the literature, like convolutional encoders (Kalchbrenner and Blunsom, 2013; Kalchbrenner et al., 2016; Gehring et al., 2017) and self-attention based encoders (Vaswani et al., 2017). We will first focus on the more common RNN-based ones:

- **Uni-Directional RNN Encoder** In the first work which successfully applied NMT in a relatively large scale scenario (Sutskever et al., 2014b), the encoder was modeled using an LSTM, which was fed with the word embeddings of the input sentence. In this case the size of each annotation vector $D_{encoder}$ is the size of the LSTM output.

- **Bi-Directional RNN Encoder** Another variation of an RNN-based encoder is the bidirectional RNN encoder Schuster and Paliwal (1997) which consists of two Uni-directional RNN encoders, one fed with the input embeddings in a left-to-right order (Forward RNN) and the other in a right-to-left order (Backward RNN). The outputs from each RNN for every input embedding are then concatenated, resulting in a representation which captures both the right and the left context of every token in the sentence. In this case the size of each annotation vector $D_{encoder}$ is the sum of the size of the forward RNN output and of the backward RNN output.

2.2.4 Decoder

Given the annotation vectors produced by the encoder, one approach (Sutskever et al., 2014b) is to use the last annotation vector $\hat{f}_{|F|}$ (generated after feeding the embedding of the last token in the input sentence) as the representation for the entire sentence. We can then "decode" the translation of the source sentence in a left-to-right manner using another neural-network component, called the "Decoder". Generally, we can say that the decoder is a parameterized function g, mapping from an annotation vector and a sequence of output embeddings to

the output distribution at a given time step:

$$p(e_t|f_{1:|F|}, e_{<t}, \theta) = g(\hat{\boldsymbol{f}}, \boldsymbol{e}_{<t}, \theta) \tag{2.6}$$

where g is a parameterized non-linear function.

While here, like in the encoder, there are many possible variations, we begin by describing the most common, RNN-based decoder. In the simplest case, the decoder is essentially an RNN-based language model: In each step, it is fed with the embedding of the last predicted word, and predicts a distribution over the possible output symbols from which we can choose or sample the next output symbol. The only difference from a vanilla RNN language model is that it is initialized or fed with the last annotation vector $\hat{\boldsymbol{f}}_{|F|}$ produced by the encoder. Specifically, the decoder may include an RNN which is initialized with $\hat{\boldsymbol{f}}_{|F|}$, whose input is the output word embedding which corresponds to the previously predicted word e_{t-1} and whose output is $\boldsymbol{s}_t \in \mathbb{R}^{D_{output}}$:

$$\boldsymbol{s}_t = \text{RNN}(\boldsymbol{s}_{t-1}, \boldsymbol{e}_t, \hat{\boldsymbol{f}}_{|\boldsymbol{F}|}) \tag{2.7}$$

We will then project $\boldsymbol{s}_t$ to the size of the output vocabulary $|\Sigma_{output}|$ using a learned projection matrix $\boldsymbol{W} \in \mathbb{R}^{D_{output} \times |\Sigma_{output}|}$, and apply the softmax function ($\boldsymbol{b} \in |\Sigma_{output}|$ is a bias parameter):

$$g(\hat{\boldsymbol{f}}, \boldsymbol{e}_{<t}, \theta) = \text{softmax}(\boldsymbol{s_t W} + \boldsymbol{b}) \tag{2.8}$$

where $\text{softmax}(\boldsymbol{x})$ is defined as:

$$\text{softmax}(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_{j=1}^{|\boldsymbol{x}|} e^{\boldsymbol{x}_j}}, \text{softmax} : \mathbb{R}^k \rightarrow \left\{ \boldsymbol{z} \in \mathbb{R}^k, z_i > 0, \sum_{i=1}^{k} z_i = 1 \right\} \tag{2.9}$$

given the above, we can estimate the probability of a translation given the

input sentence and the network's parameters using the chain rule:

$$p(E|F,\theta) = p(e_1, e_2, e_3, \ldots, e_{|E|}|f_1, f_2, f_3, \ldots, f_{|F|}) = \prod_{t=1}^{|E|} p(e_t|f_{1:|F|}, e_{<t}, \theta) \quad (2.10)$$

2.2.5 The Attention Mechanism

While relatively simple, the above approach creates an "information bottleneck" since both long and short sentences need to be "compressed" into a single fixed-sized encoding vector. To alleviate this, the attention mechanism was introduced (Bahdanau et al., 2015), proposing a new approach for using the annotation vectors produced by the encoder. Instead of using only the last annotation vector $\hat{\boldsymbol{f}}_{|F|}$ in the decoder, we compute at each time step t a new context vector $\boldsymbol{c}_t$, which is a weighted sum over all of the inputs annotation vectors:

$$\boldsymbol{c}_t = \sum_{i=1}^{|F|} \alpha_{ti} \hat{\boldsymbol{f}}_i, \boldsymbol{c}_t \in \mathbb{R}^{D_{output}} \quad (2.11)$$

where:

$$\sum_{i=1}^{|F|} \alpha_{ti} = 1 \quad (2.12)$$

since each α_{ti} value is a result of a softmax function over $|F|$ e_{ti} values. An important question in the attention mechanism is how to compute those e_{ti} values, or attention weights, which are scalars that can be interpreted as the importance of the annotation vector $\hat{\boldsymbol{f}}_i$ in the prediction of the output distribution in time step t. We will mention two popular methods here:

- **Feed Forward Attention** also known as concat attention or MLP attention. In this case the e_{ti} values are computed using a feedforward network[1]:

$$e_{ti} = a(\boldsymbol{s}_t, \hat{\boldsymbol{f}}_i) = \boldsymbol{v}_a{}^\top \tanh(\boldsymbol{W}_a[\boldsymbol{s_t}, \hat{\boldsymbol{f}}_i]) \quad (2.13)$$

[1] In the original paper ((Bahdanau et al., 2015)) the attention weights were computed using s_{t-1}. We show the version of Loung et al. (Luong et al., 2015b) which uses $\boldsymbol{s}_t$ instead.

- **Dot-Product Attention** here the e_{ti} vectors are computed using a dot product between the decoder state and the annotation vector:

$$e_{ti} = a(\boldsymbol{s}_t, \hat{\boldsymbol{f}}_i) = \boldsymbol{s}_t^\top \hat{\boldsymbol{f}}_i \tag{2.14}$$

We then use this context vector $\boldsymbol{c}_t$ when computing the output distribution in each time step. For example in Luong et al. (2015b) this was done by:

$$p(e_t|f_{1:|F|}, e_{<t}, \theta) = g(\hat{\boldsymbol{f}}, \boldsymbol{e}_{<t}, \boldsymbol{c}_t, \theta) = \text{softmax}(\boldsymbol{W}_s \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t, \boldsymbol{s}_t])) \tag{2.15}$$

Where $\boldsymbol{W}_s \in \mathbb{R}^{|\Sigma_{output}| \times 2D_{output}}$, $\boldsymbol{W}_c \in \mathbb{R}^{2D_{output} \times 2D_{output}}$ are model parameters.

2.2.6 Training Objective

In order to train an NMT system, the most popular objective is cross-entropy (a.k.a. negative log-likelihood), where we try to minimize the following term:

$$\mathcal{L} = \sum_{(f,e) \in train} \sum_{e_t \in e} -\log p(e_t|f_{1:|F|}, e_{<t}, \theta) \tag{2.16}$$

Note that this is usually optimized using Stochastic Gradient Descent (SGD), where we take the derivative of the objective w.r.t. the model parameters and update them by taking a step in the direction of the derivative. Another thing to note is that in SGD we do not compute the objective over the entire training set, but over a single example or a mini-batch.

2.2.7 Inference and Beam-search

Once we have a trained model as described above, how do we predict translations for a given source-language sentence using this model? As we noted in

the beginning of this section, our goal is to find:

$$\hat{E} = \arg\max_{E} p(E|F, \theta) \tag{2.17}$$

or according to our modeling:

$$\hat{E} = \arg\max_{E} \prod_{t=1}^{|E|} p(e_t|f_{1:|F|}, e_{<t}, \theta) \tag{2.18}$$

note that this arg max operation requires a search over an exponentially large space, as for each position in the output sequence we have $|\Sigma_{output}|$ options to choose from ($O(|\Sigma_{output}|^n)$ where n is the maximum sentence length.) Since searching over this entire space is intractable, heuristic search methods like Greedy-search or Beam-search are used.

In Greedy-search, we will choose the most probable word in each time step according to our model. While this method is relatively fast as it decodes in $O(|\Sigma_{output}| \cdot n)$, it may result in sub-optimal translations since an early word-choice error may lead us to more mistakes down the line (as the next words depend on the previously selected words - an "auto-regressive" model).

A common approach to alleviate this issue is Beam-search. In Beam-search, we begin with an empty hypothesis and extend it by choosing the K words which will lead us to the K-highest scoring partial hypotheses. Then, in the next time step, we will expand each of these K-best hypotheses and again choose a new set of K-best partial hypotheses. We will continue iterating this process until all the K-best hypotheses in the end of the iteration are complete hypotheses (i.e. hypotheses that end in an end-of-sequence symbol). This K constant is called the "beam size". See Figure 2.2 for an example of a beam search procedure from an NMT system.[2]

[2]This figure was generated using the Nematus NMT toolkit (Sennrich et al., 2017).

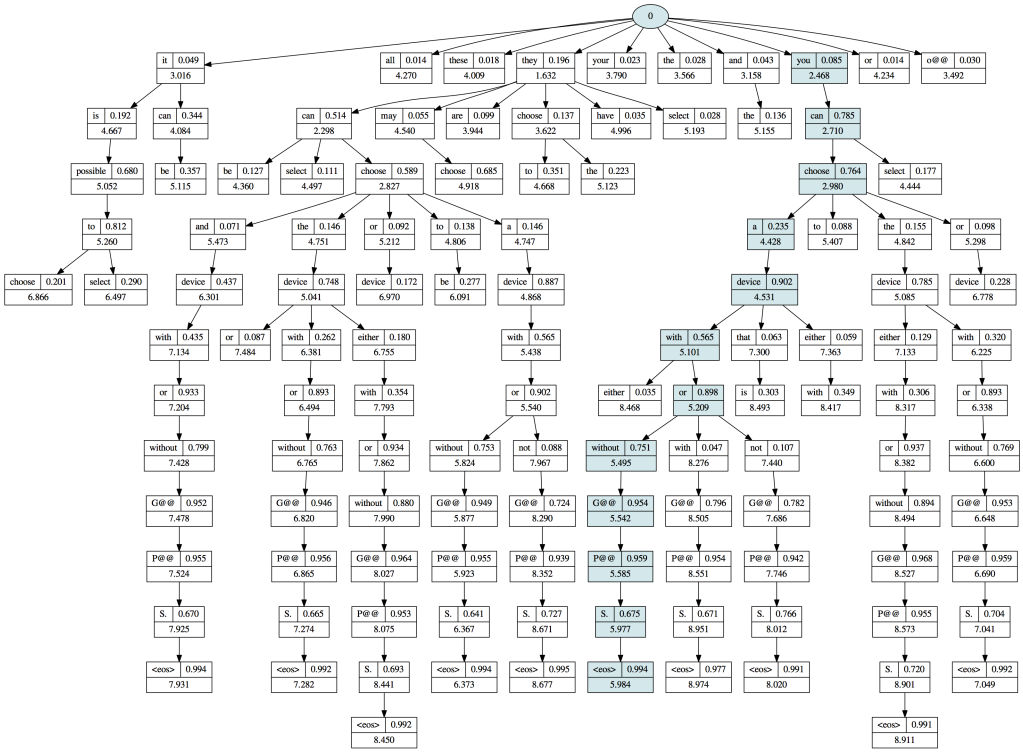


Figure 2.2: Example Beam-search graph from a German-to-English NMT system, where the beam size is set to 10. Each node corresponds to a partial hypothesis, and leaves correspond to complete hypotheses. Nodes in light blue are participating in the highest-scored complete hypothesis. The number to the right of each word is the probability of the word according to the local softmax computation in this time step. The numbers beneath each word is the negative log-likelihood of the partial hypothesis that ends in this word.

Chapter 3

Hard Attention Architectures for Morphological Inflection Generation

Morphological Inflection Generation with Hard Monotonic Attention

Roee Aharoni & Yoav Goldberg
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{roee.aharoni,yoav.goldberg}@gmail.com

Abstract

We present a neural model for morphological inflection generation which employs a hard attention mechanism, inspired by the nearly-monotonic alignment commonly found between the characters in a word and the characters in its inflection. We evaluate the model on three previously studied morphological inflection generation datasets and show that it provides state of the art results in various setups compared to previous neural and non-neural approaches. Finally we present an analysis of the continuous representations learned by both the hard and soft attention (Bahdanau et al., 2015) models for the task, shedding some light on the features such models extract.

1 Introduction

Morphological inflection generation involves generating a target word (e.g. "härtestem", the German word for "hardest"), given a source word (e.g. "hart", the German word for "hard") and the morpho-syntactic attributes of the target (POS=adjective, gender=masculine, type=superlative, etc.).

The task is important for many down-stream NLP tasks such as machine translation, especially for dealing with data sparsity in morphologically rich languages where a lemma can be inflected into many different word forms. Several studies have shown that translating into lemmas in the target language and then applying inflection generation as a post-processing step is beneficial for phrase-based machine translation (Minkov et al., 2007; Toutanova et al., 2008; Clifton and Sarkar, 2011; Fraser et al., 2012; Chahuneau et al., 2013) and more recently for neural machine translation (García-Martínez et al., 2016).

The task was traditionally tackled with hand engineered finite state transducers (FST) (Koskenniemi, 1983; Kaplan and Kay, 1994) which rely on expert knowledge, or using trainable weighted finite state transducers (Mohri et al., 1997; Eisner, 2002) which combine expert knowledge with data-driven parameter tuning. Many other machine-learning based methods (Yarowsky and Wicentowski, 2000; Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Hulden et al., 2014; Ahlberg et al., 2015; Nicolai et al., 2015) were proposed for the task, although with specific assumptions about the set of possible processes that are needed to create the output sequence.

More recently, the task was modeled as neural sequence-to-sequence learning over character sequences with impressive results (Faruqui et al., 2016). The vanilla encoder-decoder models as used by Faruqui et al. compress the input sequence to a single, fixed-sized continuous representation. Instead, the soft-attention based sequence to sequence learning paradigm (Bahdanau et al., 2015) allows directly conditioning on the entire input sequence representation, and was utilized for morphological inflection generation with great success (Kann and Schütze, 2016b,a).

However, the neural sequence-to-sequence models require large training sets in order to perform well: their performance on the relatively small CELEX dataset is inferior to the latent variable WFST model of Dreyer et al. (2008). Interestingly, the neural WFST model by Rastogi et al. (2016) also suffered from the same issue on the CELEX dataset, and surpassed the latent variable model only when given twice as much data to train on.

We propose a model which handles the above issues by directly modeling an almost monotonic

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 2004–2015
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-1183

alignment between the input and output character sequences, which is commonly found in the morphological inflection generation task (e.g. in languages with concatenative morphology). The model consists of an encoder-decoder neural network with a dedicated control mechanism: in each step, the model attends to a *single* input state and either writes a symbol to the output sequence or advances the attention pointer to the next state from the bi-directionally encoded sequence, as described visually in Figure 1.

This modeling suits the natural monotonic alignment between the input and output, as the network learns to attend to the relevant inputs before writing the output which they are aligned to. The encoder is a bi-directional RNN, where each character in the input word is represented using a concatenation of a forward RNN and a backward RNN states over the word's characters. The combination of the bi-directional encoder and the controllable hard attention mechanism enables to condition the output on the entire input sequence. Moreover, since each character representation is aware of the neighboring characters, non-monotone relations are also captured, which is important in cases where segments in the output word are a result of long range dependencies in the input word. The recurrent nature of the decoder, together with a dedicated feedback connection that passes the last prediction to the next decoder step explicitly, enables the model to also condition the current output on all the previous outputs at each prediction step.

The hard attention mechanism allows the network to jointly align and transduce while using a focused representation at each step, rather then the weighted sum of representations used in the soft attention model. This makes our model Resolution Preserving (Kalchbrenner et al., 2016) while also keeping decoding time linear in the output sequence length rather than multiplicative in the input and output lengths as in the soft-attention model. In contrast to previous sequence-to-sequence work, we do not require the training procedure to also learn the alignment. Instead, we use a simple training procedure which relies on independently learned character-level alignments, from which we derive gold transduction+control sequences. The network can then be trained using straightforward cross-entropy loss.

To evaluate our model, we perform extensive experiments on three previously studied morphological inflection generation datasets: the CELEX dataset (Baayen et al., 1993), the Wiktionary dataset (Durrett and DeNero, 2013) and the SIGMORPHON2016 dataset (Cotterell et al., 2016). We show that while our model is on par with or better than the previous neural and non-neural state-of-the-art approaches, it also performs significantly better with very small training sets, being the first neural model to surpass the performance of the weighted FST model with latent variables which was specifically tailored for the task by Dreyer et al. (2008). Finally, we analyze and compare our model and the soft attention model, showing how they function very similarly with respect to the alignments and representations they learn, in spite of our model being much simpler. This analysis also sheds light on the representations such models learn for the morphological inflection generation task, showing how they encode specific features like a symbol's type and the symbol's location in a sequence.

To summarize, our contributions in this paper are three-fold:

1. We present a hard attention model for nearly-monotonic sequence to sequence learning, as common in the morphological inflection setting.

2. We evaluate the model on the task of morphological inflection generation, establishing a new state of the art on three previously-studied datasets for the task.

3. We perform an analysis and comparison of our model and the soft-attention model, shedding light on the features such models extract for the inflection generation task.

2 The Hard Attention Model

2.1 Motivation

We would like to transduce an input sequence, $x_{1:n} \in \Sigma_x^*$ into an output sequence, $y_{1:m} \in \Sigma_y^*$, where Σ_x and Σ_y are the input and output vocabularies, respectively. Imagine a machine with read-only random access to the encoding of the input sequence, and a single pointer that determines the current read location. We can then model sequence transduction as a series of pointer movement and write operations. If we assume the alignment is monotone, the machine can be simpli-

fied: the memory can be read in sequential order, where the pointer movement is controlled by a single "move forward" operation (step) which we add to the output vocabulary. We implement this behavior using an encoder-decoder neural network, with a control mechanism which determines in each step of the decoder whether to write an output symbol or promote the attention pointer the next element of the encoded input.

2.2 Model Definition

In prediction time, we seek the output sequence $y_{1:m} \in \Sigma_y^*$, for which:

$$y_{1:m} = \arg\max_{y' \in \Sigma_y^*} p(y'|x_{1:n}, f) \quad (1)$$

Where $x \in \Sigma_x^*$ is the input sequence and $f = \{f_1, \ldots, f_l\}$ is a set of attributes influencing the transduction task (in the inflection generation task these would be the desired morpho-syntactic attributes of the output sequence). Given a nearly-monotonic alignment between the input and the output, we replace the search for a sequence of letters with a sequence of actions $s_{1:q} \in \Sigma_s^*$, where $\Sigma_s = \Sigma_y \cup \{step\}$. This sequence is a series of step and write actions required to go from $x_{1:n}$ to $y_{1:m}$ according to the monotonic alignment between them (we will elaborate on the deterministic process of getting $s_{1:q}$ from a monotonic alignment between $x_{1:n}$ to $y_{1:m}$ in section 2.4). In this case we define: [1]

$$\begin{aligned} s_{1:q} &= \arg\max_{s' \in \Sigma_s^*} p(s'|x_{1:n}, f) \\ &= \arg\max_{s' \in \Sigma_s^*} \prod_{s'_i \in s'} p(s'_i|s'_1 \ldots s'_{i-1}, x_{1:n}, f) \end{aligned} \quad (2)$$

which we can estimate using a neural network:

$$s_{1:q} = \arg\max_{s' \in \Sigma_s^*} \text{NN}(x_{1:n}, f, \Theta) \quad (3)$$

where the network's parameters Θ are learned using a set of training examples. We will now describe the network architecture.

[1] We note that our model (Eq. 2) solves a different objective than (Eq 1), as it searches for the *best derivation* and not *the best sequence*. In order to accurately solve (1) we would need to marginalize over the different derivations leading to the same sequence, which is computationally challenging. However, as we see in the experiments section, the best-derivation approximation is effective in practice.

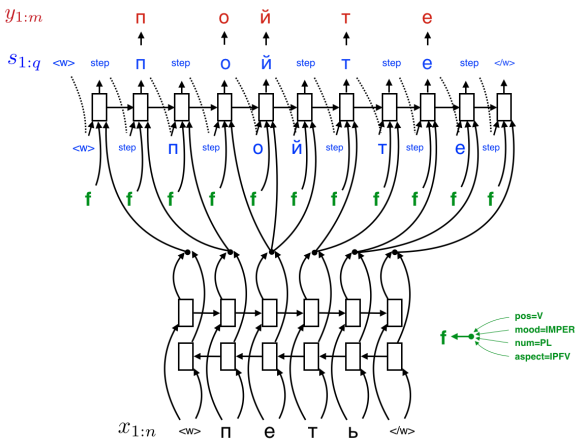


Figure 1: The hard attention network architecture. A round tip expresses concatenation of the inputs it receives. The attention is promoted to the next input element once a step action is predicted.

2.3 Network Architecture

Notation We use bold letters for vectors and matrices. We treat LSTM as a parameterized function $\mathbf{LSTM}_\theta(\mathbf{x}_1 \ldots \mathbf{x}_n)$ mapping a sequence of input vectors $\mathbf{x}_1 \ldots \mathbf{x}_n$ to a an output vector $\mathbf{h}_n$. The equations for the LSTM variant we use are detailed in the supplementary material of this paper.

Encoder For every element in the input sequence: $x_{1:n} = x_1 \ldots x_n$, we take the corresponding embedding: $\mathbf{e}_{x_1} \ldots \mathbf{e}_{x_n}$, where: $\mathbf{e}_{x_i} \in \mathbb{R}^E$. These embeddings are parameters of the model which will be learned during training. We then feed the embeddings into a bi-directional LSTM encoder (Graves and Schmidhuber, 2005) which results in a sequence of vectors: $\mathbf{x}_{1:n} = \mathbf{x}_1 \ldots \mathbf{x}_n$, where each vector $\mathbf{x}_i \in \mathbb{R}^{2H}$ is a concatenation of: $\mathbf{LSTM}_{\mathbf{forward}}(\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, \ldots \mathbf{e}_{x_i})$ and $\mathbf{LSTM}_{\mathbf{backward}}(\mathbf{e}_{x_n}, \mathbf{e}_{x_{n-1}} \ldots \mathbf{e}_{x_i})$, the forward LSTM and the backward LSTM outputs when fed with $\mathbf{e}_{x_i}$.

Decoder Once the input sequence is encoded, we feed the decoder RNN, $\mathbf{LSTM}_{\mathbf{dec}}$, with three inputs at each step:

1. The current attended input, $\mathbf{x}_a \in \mathbb{R}^{2H}$, initialized with the first element of the encoded sequence, $\mathbf{x}_1$.
2. A set of embeddings for the attributes that influence the generation process, concatenated to a single vector: $\mathbf{f} = [\mathbf{f}_1 \ldots \mathbf{f}_l] \in \mathbb{R}^{F \cdot l}$.
3. $\mathbf{s}_{i-1} \in \mathbb{R}^E$, which is an embedding for the

predicted output symbol in the previous decoder step.

Those three inputs are concatenated into a single vector $\mathbf{z}_i = [\mathbf{x}_a, \mathbf{f}, \mathbf{s}_{i-1}] \in \mathbb{R}^{2H+F\cdot l+E}$, which is fed into the decoder, providing the decoder output vector: $\mathbf{LSTM_{dec}}(\mathbf{z}_1 \dots \mathbf{z}_i) \in \mathbb{R}^H$. Finally, to model the distribution over the possible actions, we project the decoder output to a vector of $|\Sigma_s|$ elements, followed by a softmax layer:

$$p(s_i = c) = softmax_c(\mathbf{W} \cdot \mathbf{LSTM_{dec}}(\mathbf{z}_1 \dots \mathbf{z}_i) + \mathbf{b}) \quad (4)$$

Control Mechanism When the most probable action is *step*, the attention is promoted so $\mathbf{x}_a$ contains the next encoded input representation to be used in the next step of the decoder. The process is demonstrated visually in Figure 1.

2.4 Training the Model

For every example: $(x_{1:n}, y_{1:m}, f)$ in the training data, we should produce a sequence of step and write actions $s_{1:q}$ to be predicted by the decoder. The sequence is dependent on the alignment between the input and the output: ideally, the network will attend to all the input characters aligned to an output character before writing it. While recent work in sequence transduction advocate jointly training the alignment and the decoding mechanisms (Bahdanau et al., 2015; Yu et al., 2016), we instead show that in our case it is worthwhile to decouple these stages and learn a hard alignment beforehand, using it to guide the training of the encoder-decoder network and enabling the use of correct alignments for the attention mechanism from the beginning of the network training phase. Thus, our training procedure consists of three stages: learning hard alignments, deriving oracle actions from the alignments, and learning a neural transduction model given the oracle actions.

Learning Hard Alignments We use the character alignment model of Sudoh et al. (2013), based on a Chinese Restaurant Process which weights single alignments (character-to-character) in proportion to how many times such an alignment has been seen elsewhere out of all possible alignments. The aligner implementation we used produces either 0-to-1, 1-to-0 or 1-to-1 alignments.

Deriving Oracle Actions We infer the sequence of actions $s_{1:q}$ from the alignments by the deterministic procedure described in Algorithm 1. An example of an alignment with the resulting oracle action sequence is shown in Figure 2, where a_4 is a 0-to-1 alignment and the rest are 1-to-1 alignments.

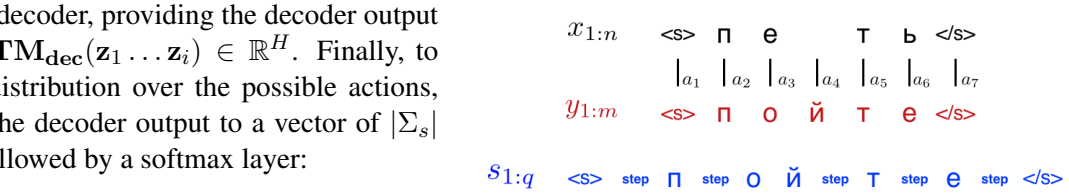


Figure 2: Top: an alignment between a lemma $x_{1:n}$ and an inflection $y_{1:m}$ as predicted by the aligner. Bottom: $s_{1:q}$, the sequence of actions to be predicted by the network, as produced by Algorithm 1 for the given alignment.

Algorithm 1 Generates the oracle action sequence $s_{1:q}$ from the alignment between $x_{1:n}$ and $y_{1:m}$

Require: a, the list of either 1-to-1, 1-to-0 or 0-to-1 alignments between $x_{1:n}$ and $y_{1:m}$

```
Initialize s as an empty sequence
for each a_i = (x_{a_i}, y_{a_i}) ∈ a do
    if a_i is a 1-to-0 alignment then
        s.append(step)
    else
        s.append(y_{a_i})
        if a_{i+1} is not a 0-to-1 alignment then
        s.append(step)
return s
```

This procedure makes sure that all the source input elements aligned to an output element are read (using the step action) before writing it.

Learning a Neural Transduction Model The network is trained to mimic the actions of the oracle, and at inference time, it will predict the actions according to the input. We train it using a conventional cross-entropy loss function per example:

$$\mathcal{L}(x_{1:n}, y_{1:m}, f, \Theta) = -\sum_{s_j \in s_{1:q}} \log softmax_{s_j}(\mathbf{d}),$$
$$\mathbf{d} = \mathbf{W} \cdot \mathbf{LSTM_{dec}}(\mathbf{z}_1 \dots \mathbf{z}_i) + \mathbf{b} \quad (5)$$

Transition System An alternative view of our model is that of a *transition system* with ADVANCE and WRITE(CH) actions, where the oracle is derived from a given hard alignment, the input is encoded using a biRNN, and the next action is determined by an RNN over the previous inputs and actions.

3 Experiments

We perform extensive experiments with three previously studied morphological inflection generation datasets to evaluate our hard attention model in various settings. In all experiments we compare our hard attention model to the best performing neural and non-neural models which were previously published on those datasets, and to our implementation of the global (soft) attention model presented by Luong et al. (2015) which we train with identical hyper-parameters as our hard-attention model. The implementation details for our models are described in the supplementary material section of this paper. The source code and data for our models is available on github.[2]

CELEX Our first evaluation is on a very small dataset, to see if our model indeed avoids the tendency to overfit with small training sets. We report exact match accuracy on the German inflection generation dataset compiled by Dreyer et al. (2008) from the CELEX database (Baayen et al., 1993). The dataset includes only 500 training examples for each of the four inflection types: 13SIA→13SKE, 2PIE→13PKE, 2PKE→z, and rP→pA which we refer to as 13SIA, 2PIE, 2PKE and rP, respectively.[3] We first compare our model to three competitive models from the literature that reported results on this dataset: the Morphological Encoder-Decoder (MED) of Kann and Schütze (2016a) which is based on the soft-attention model of Bahdanau et al. (2015), the neural-weighted FST of Rastogi et al. (2016) which uses stacked bi-directional LSTM's to weigh its arcs (NWFST), and the model of Dreyer et al. (2008) which uses a weighted FST with latent-variables structured particularly for morphological string transduction tasks (LAT). Following previous reports on this dataset, we use the same data splits as Dreyer et al. (2008), dividing the data for each inflection type into five folds, each consisting of 500 training, 1000 development and 1000 test examples. We train a separate model for each fold and report exact match accuracy, averaged over the five folds.

Wiktionary To neutralize the negative effect of very small training sets on the performance of the different learning approaches, we also evaluate our model on the dataset created by Durrett and DeNero (2013), which contains up to 360k training examples per language. It was built by extracting Finnish, German and Spanish inflection tables from Wiktionary, used in order to evaluate their system based on string alignments and a semi-CRF sequence classifier with linguistically inspired features, which we use a baseline. We also used the dataset expansion made by Nicolai et al. (2015) to include French and Dutch inflections as well. Their system also performs an align-and-transduce approach, extracting rules from the aligned training set and applying them in inference time with a proprietary character sequence classifier. In addition to those systems we also compare to the results of the recent neural approach of Faruqui et al. (2016), which did not use an attention mechanism, and Yu et al. (2016), which coupled the alignment and transduction tasks.

SIGMORPHON As different languages show different morphological phenomena, we also experiment with how our model copes with these various phenomena using the morphological inflection dataset from the SIGMORPHON2016 shared task (Cotterell et al., 2016). Here the training data consists of ten languages, with five morphological system types (detailed in Table 3): Russian (RU), German (DE), Spanish (ES), Georgian (GE), Finnish (FI), Turkish (TU), Arabic (AR), Navajo (NA), Hungarian (HU) and Maltese (MA) with roughly 12,800 training and 1600 development examples per language. We compare our model to two soft attention baselines on this dataset: MED (Kann and Schütze, 2016b), which was the best participating system in the shared task, and our implementation of the global (soft) attention model presented by Luong et al. (2015).

4 Results

In all experiments, for both the hard and soft attention models we implemented, we report results using an ensemble of 5 models with different random initializations by using majority voting on the final sequences the models predicted, as proposed by Kann and Schütze (2016a). This was done to perform fair comparison to the models of Kann and Schütze (2016a,b); Faruqui et al. (2016) which we compare to, that also perform a similar ensem-

[2] https://github.com/roeeaharoni/morphological-reinflection

[3] The acronyms stand for: 13SIA=1st/3rd person, singular, indefinite, past;13SKE=1st/3rd person, subjunctive, present; 2PIE=2nd person, plural, indefinite, present;13PKE=1st/3rd person, plural, subjunctive, present; 2PKE=2nd person, plural, subjunctive, present; z=infinitive; rP=imperative, plural; pA=past participle.

	13SIA	2PIE	2PKE	rP	Avg.
MED (Kann and Schütze, 2016a)	83.9	95	87.6	84	87.62
NWFST (Rastogi et al., 2016)	86.8	94.8	87.9	81.1	87.65
LAT (Dreyer et al., 2008)	**87.5**	93.4	87.4	84.9	88.3
Soft	83.1	93.8	88	83.2	87
Hard	85.8	**95.1**	**89.5**	**87.2**	**89.44**

Table 1: Results on the CELEX dataset

	DE-N	DE-V	ES-V	FI-NA	FI-V	FR-V	NL-V	Avg.
Durrett and DeNero (2013)	88.31	94.76	99.61	92.14	97.23	98.80	90.50	94.47
Nicolai et al. (2015)	88.6	97.50	99.80	93.00	**98.10**	**99.20**	96.10	96.04
Faruqui et al. (2016)	88.12	**97.72**	**99.81**	95.44	97.81	98.82	96.71	96.34
Yu et al. (2016)	87.5	92.11	99.52	95.48	**98.10**	98.65	95.90	95.32
Soft	88.18	95.62	99.73	93.16	97.74	98.79	96.73	95.7
Hard	**88.87**	97.35	99.79	**95.75**	98.07	99.04	**97.03**	**96.55**

Table 2: Results on the Wiktionary datasets

	suffixing+stem changes			circ.	suffixing+agg.+v.h.			c.h.	templatic		
	RU	DE	ES	GE	FI	TU	HU	NA	AR	MA	Avg.
MED	91.46	95.8	98.84	98.5	95.47	98.93	96.8	91.48	**99.3**	**88.99**	95.56
Soft	92.18	96.51	98.88	**98.88**	**96.99**	**99.37**	**97.01**	**95.41**	**99.3**	88.86	**96.34**
Hard	**92.21**	**96.58**	**98.92**	98.12	95.91	97.99	96.25	93.01	98.77	88.32	95.61

Table 3: Results on the SIGMORPHON 2016 morphological inflection dataset. The text above each language lists the morphological phenomena it includes: circ.=circumfixing, agg.=agglutinative, v.h.=vowel harmony, c.h.=consonant harmony

bling technique.

On the low resource setting (CELEX), our hard attention model significantly outperforms both the recent neural models of Kann and Schütze (2016a) (MED) and Rastogi et al. (2016) (NWFST) and the morphologically aware latent variable model of Dreyer et al. (2008) (LAT), as detailed in Table 1. In addition, it significantly outperforms our implementation of the soft attention model (Soft). It is also, to our knowledge, the first model that surpassed in overall accuracy the latent variable model on this dataset. We attribute our advantage over the soft attention models to the ability of the hard attention control mechanism to harness the monotonic alignments found in the data. The advantage over the FST models may be explained by our conditioning on the entire output history which is not available in those models. Figure 3 plots the train-set and dev-set accuracies of the soft and hard attention models as a function of the training epoch. While both models perform similarly on the train-set (with the soft attention model fitting it slightly faster), the hard attention model performs significantly better on the dev-set. This shows the soft attention model's tendency to overfit on the small dataset, as it is not enforcing the monotonic assumption of the hard attention model.

On the large training set experiments (Wiktionary), our model is the best performing model on German verbs, Finnish nouns/adjectives and Dutch verbs, resulting in the highest reported average accuracy across all inflection types when compared to the four previous neural and non-neural state of the art baselines, as detailed in Table 2. This shows the robustness of our model also with large amounts of training examples, and the advantage the hard attention mechanism provides over the encoder-decoder approach of Faruqui et al. (2016) which does not employ an attention mechanism. Our model is also significantly more accurate than the model of Yu et al. (2016), which shows the advantage of using independently learned alignments to guide the network's attention from the beginning of the training process. While our soft-attention implementation outperformed the models of Yu et al. (2016) and Durrett and DeNero (2013), it still performed inferiorly to the hard attention model.

As can be seen in Table 3, on the SIGMORPHON 2016 dataset our model performs

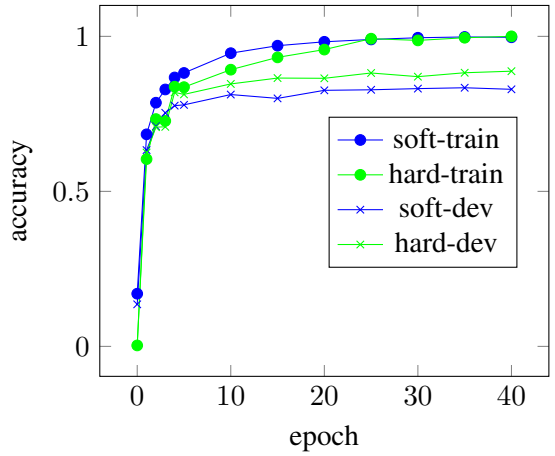


Figure 3: Learning curves for the soft and hard attention models on the first fold of the CELEX dataset

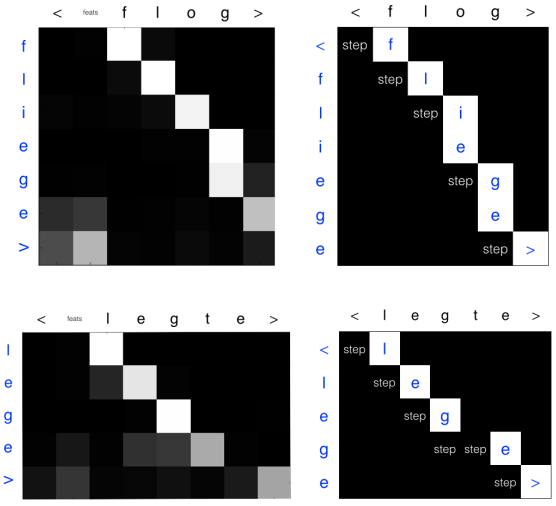


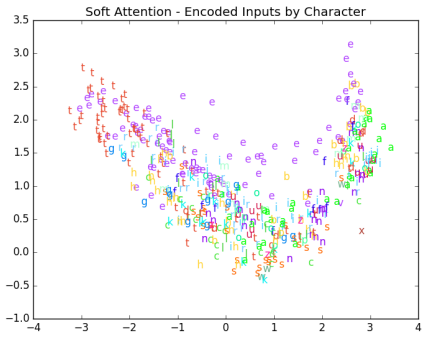
Figure 4: A comparison of the alignments as predicted by the soft attention (left) and the hard attention (right) models on examples from CELEX.

better than both soft-attention baselines for the suffixing+stem-change languages (Russian, German and Spanish) and is slightly less accurate than our implementation of the soft attention model on the rest of the languages, which is now the best performing model on this dataset to our knowledge. We explain this by looking at the languages from a linguistic typology point of view, as detailed in Cotterell et al. (2016). Since Russian, German and Spanish employ a suffixing morphology with internal stem changes, they are more suitable for monotonic alignment as the transformations they need to model are the addition of suffixes and changing characters in the stem. The rest of the languages in the dataset employ more context sensitive morphological phenomena like vowel harmony and consonant harmony, which require to model long range dependencies in the input sequence which better suits the soft attention mechanism. While our implementation of the soft attention model and MED are very similar modelwise, we hypothesize that our soft attention model results are better due to the fact that we trained the model for 100 epochs and picked the best performing model on the development set, while the MED system was trained for a fixed amount of 20 epochs (although trained on more data – both train and development sets).

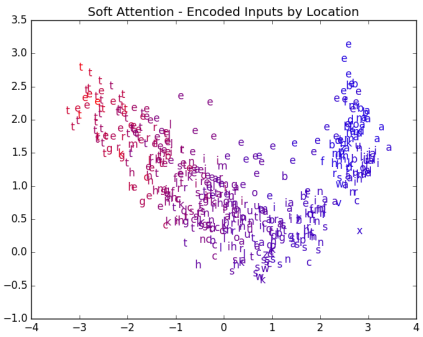
5 Analysis

The Learned Alignments In order to see if the alignments predicted by our model fit the monotonic alignment structure found in the data, and whether are they more suitable for the task when compared to the alignments found by the soft attention model, we examined alignment predictions of the two models on examples from the development portion of the CELEX dataset, as depicted in Figure 4. First, we notice the alignments found by the soft attention model are also monotonic, supporting our modeling approach for the task. Figure 4 (bottom-right) also shows how the hard-attention model performs deletion (legte→lege) by predicting a sequence of two step operations. Another notable morphological transformation is the one-to-many alignment, found in the top example: flog→fliege, where the model needs to transform a character in the input, *o*, to two characters in the output, *ie*. This is performed by two consecutive *write* operations after the *step* operation of the relevant character to be replaced. Notice that in this case, the soft attention model performs a different alignment by aligning the character *i* to *o* and the character *g* to the sequence *eg*, which is not the expected alignment in this case from a linguistic point of view.

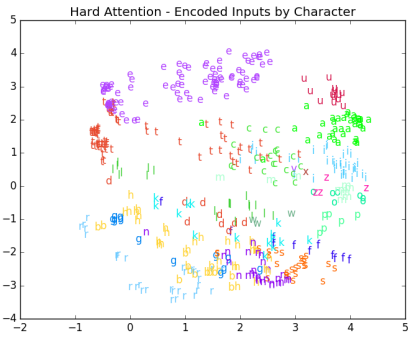
The Learned Representations How does the soft-attention model manage to learn nearly-perfect monotonic alignments? Perhaps the the network learns to encode the sequential position as part of its encoding of an input element? More generally, what information is encoded by the soft and hard alignment encoders? We selected 500 random encoded characters-in-context from input



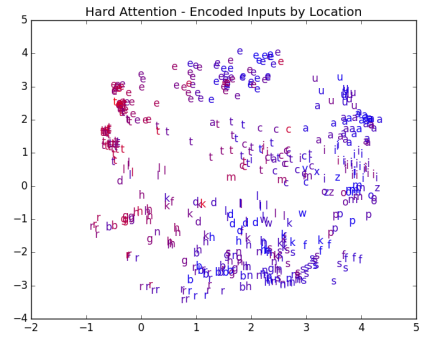
(a) Colors indicate which character is encoded.



(c) Colors indicate the character's position.



(b) Colors indicate which character is encoded.



(d) Colors indicate the character's position.

Figure 5: SVD dimension reduction to 2D of 500 character representations in context from the encoder, for both the soft attention (top) and hard attention (bottom) models.

words in the CELEX development set, where every encoded representation is a vector in $\mathbb{R}^{200}$. Since those vectors are outputs from the bi-LSTM encoders of the models, every vector of this form carries information of the specific character with its entire context. We project these encodings into 2-D using SVD and plot them twice, each time using a different coloring scheme. We first color each point according to the character it represents (Figures 5a, 5b). In the second coloring scheme (Figures 5c, 5d), each point is colored according to the character's sequential position in the word it came from, blue indicating positions near the beginning of the word, and red positions near its end.

While both models tend to cluster representations for similar characters together (Figures 5a, 5b), the hard attention model tends to have much more isolated character clusters. Figures 5c, 5d show that both models also tend to learn representations which are sensitive to the position of the character, although it seems that here the soft attention model is more sensitive to this information as its coloring forms a nearly-perfect red-to-blue transition on the X axis. This may be explained by the soft-attention mechanism encouraging the encoder to encode positional information in the input representations, which may help it to predict better attention scores, and to avoid collisions when computing the weighted sum of representations for the context vector. In contrast, our hard-attention model has other means of obtaining the position information in the decoder using the step actions, and for that reason it does not encode it as strongly in the representations of the inputs. This behavior may allow it to perform well even with fewer examples, as the location information is represented more explicitly in the model using the step actions.

6 Related Work

Many previous works on inflection generation used machine learning methods (Yarowsky and Wicentowski, 2000; Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Hulden et al., 2014; Ahlberg et al., 2015; Nicolai et al., 2015) with assumptions about the set of possible processes needed to create the output word. Our work was mainly inspired by Faruqui et al. (2016) which trained an independent encoder-decoder neural

network for every inflection type in the training data, alleviating the need for feature engineering. Kann and Schütze (2016b,a) tackled the task with a *single* soft attention model (Bahdanau et al., 2015) for all inflection types, which resulted in the best submission at the SIGMORPHON 2016 shared task (Cotterell et al., 2016). In another closely related work, Rastogi et al. (2016) model the task with a WFST in which the arc weights are learned by optimizing a global loss function over all the possible paths in the state graph, while modeling contextual features with bi-directional LSTMS. This is similar to our approach, where instead of learning to mimic a single greedy alignment as we do, they sum over all possible alignments. While not committing to a single greedy alignment could in theory be beneficial, we see in Table 1 that—at least for the low resource scenario—our greedy approach is more effective in practice. Another recent work (Kann et al., 2016) proposed performing neural multi-source morphological reinflection, generating an inflection from several source forms of a word.

Previous works on neural sequence transduction include the RNN Transducer (Graves, 2012) which uses two independent RNN's over monotonically aligned sequences to predict a distribution over the possible output symbols in each step, including a null symbol to model the alignment. Yu et al. (2016) improved this by replacing the null symbol with a dedicated learned transition probability. Both models are trained using a forward-backward approach, marginalizing over all possible alignments. Our model differs from the above by learning the alignments independently, thus enabling a dependency between the encoder and decoder. While providing better results than Yu et al. (2016), this also simplifies the model training using a simple cross-entropy loss. A recent work by Raffel et al. (2017) jointly learns the hard monotonic alignments and transduction while maintaining the dependency between the encoder and the decoder. Jaitly et al. (2015) proposed the Neural Transducer model, which is also trained on external alignments. They divide the input into blocks of a constant size and perform soft attention separately on each block. Lu et al. (2016) used a combination of an RNN encoder with a CRF layer to model the dependencies in the output sequence. An interesting comparison between "traditional" sequence transduction models (Bisani and Ney, 2008; Jiampojamarn et al., 2010; Novak et al., 2012) and encoder-decoder neural networks for monotone string transduction tasks was done by Schnober et al. (2016), showing that in many cases there is no clear advantage to one approach over the other.

Regarding task-specific improvements to the attention mechanism, a line of work on attention-based speech recognition (Chorowski et al., 2015; Bahdanau et al., 2016) proposed adding location awareness by using the previous attention weights when computing the next ones, and preventing the model from attending on too many or too few inputs using "sharpening" and "smoothing" techniques on the attention weight distributions. Cohn et al. (2016) offered several changes to the attention score computation to encourage well-known modeling biases found in traditional machine translation models like word fertility, position and alignment symmetry. Regarding the utilization of independent alignment models for training attention-based networks, Mi et al. (2016) showed that the distance between the attention-infused alignments and the ones learned by an independent alignment model can be added to the networks' training objective, resulting in an improved translation and alignment quality.

7 Conclusion

We presented a hard attention model for morphological inflection generation. The model employs an explicit alignment which is used to train a neural network to perform transduction by decoding with a hard attention mechanism. Our model performs better than previous neural and non-neural approaches on various morphological inflection generation datasets, while staying competitive with dedicated models even with very few training examples. It is also computationally appealing as it enables linear time decoding while staying resolution preserving, i.e. not requiring to compress the input sequence to a single fixed-sized vector. Future work may include applying our model to other nearly-monotonic align-and-transduce tasks like abstractive summarization, transliteration or machine translation.

Acknowledgments

This work was supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI), and The Israeli Science Foundation (grant number 1555/15).

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *NAACL HLT 2015*. pages 1024–1029.

R Harald Baayen, Richard Piepenbrock, and Rijn van H. 1993. The {CELEX} lexical data base on {CD-ROM} .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations (ICLR)* .

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Yoshua Bengio, et al. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 4945–4949.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Commun.* 50(5):434–451. https://doi.org/10.1016/j.specom.2008.01.002.

Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. Translating into morphologically rich languages with synthetic phrases. In *EMNLP*. pages 1677–1687.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems 28*, pages 577–585.

Ann Clifton and Anoop Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *ACL*. pages 32–42.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 876–885. http://www.aclweb.org/anthology/N16-1102.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *EMNLP*. pages 616–627.

Markus Dreyer, Jason R Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the conference on empirical methods in natural language processing*. pages 1080–1089.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *NAACL HLT 2013*. pages 1185–1195.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th annual meeting on Association for Computational Linguistics*. pages 1–8.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *NAACL HLT 2016*.

Alexander M. Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling inflection and word-formation in smt. In *EACL*. pages 664–674.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *arXiv preprint arXiv:1609.04621* .

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18(5-6):602–610.

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

Mans Hulden, Markus Forsberg, and Malin Ahlberg. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *EACL*. pages 569–578.

Navdeep Jaitly, David Sussillo, Quoc V Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio. 2015. A neural transducer. *arXiv preprint arXiv:1511.04868* .

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training framework. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Los Angeles, California, pages 697–700. http://www.aclweb.org/anthology/N10-1103.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099* .

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural multi-source morphological reinflection. *EACL 2017* .

Katharina Kann and Hinrich Schütze. 2016a. Med: The lmu system for the sigmorphon 2016 shared task on morphological reinflection.

Katharina Kann and Hinrich Schütze. 2016b. Single-model encoder-decoder with explicit morphological representation for reinflection. In *ACL*.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3):331–378.

Kimmo Koskenniemi. 1983. Two-level morphology: A general computational model of word-form recognition and production. Technical report.

Liang Lu, Lingpeng Kong, Chris Dyer, Noah A Smith, and Steve Renals. 2016. Segmental recurrent neural networks for end-to-end speech recognition. *arXiv preprint arXiv:1603.00223* .

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2283–2288. https://aclweb.org/anthology/D16-1249.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 128–135. http://www.aclweb.org/anthology/P07-1017.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 1997. A rational design for a weighted finite-state transducer library. In *International Workshop on Implementing Automata*. pages 144–158.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *NAACL HLT 2015*. pages 922–931.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*. Association for Computational Linguistics, Donostia–San Sebastin, pages 45–49. http://www.aclweb.org/anthology/W12-6208.

C. Raffel, T. Luong, P. J. Liu, R. J. Weiss, and D. Eck. 2017. Online and Linear-Time Attention by Enforcing Monotonic Alignments. *arXiv preprint arXiv:1704.00784* .

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proc. of NAACL*.

Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. 2016. Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 1703–1714. http://aclweb.org/anthology/C16-1160.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2013. Noise-aware character alignment for bootstrapping statistical machine transliteration from bilingual corpora. In *EMNLP 2013*. pages 204–209.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *ACL*. pages 514–522.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *ACL*.

Lei Yu, Jan Buys, and Phil Blunsom. 2016. Online segment to segment neural transduction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1307–1316. https://aclweb.org/anthology/D16-1138.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Supplementary Material

Training Details, Implementation and Hyper Parameters

To train our models, we used the train portion of the datasets as-is and evaluated on the test portion the model which performed best on the development portion of the dataset, without conducting any specific pre-processing steps on the data. We train the models for a maximum of 100 epochs over the training set. To avoid long training time, we trained the model for 20 epochs for datasets larger than 50k examples, and for 5 epochs for datasets larger than 200k examples. The models were implemented using the python bindings of the dynet toolkit.[4]

We trained the network by optimizing the expected output sequence likelihood using cross-entropy loss as mentioned in equation 5. For optimization we used ADADELTA (Zeiler, 2012) without regularization. We updated the weights after every example (i.e. mini-batches of size 1). We used the dynet toolkit implementation of an LSTM network with two layers for all models, each having 100 entries in both the encoder and decoder. The character embeddings were also vectors with 100 entries for the CELEX experiments, and with 300 entries for the SIGMORPHON and Wiktionary experiments.

The morpho-syntactic attribute embeddings were vectors of 20 entries in all experiments. We did not use beam search while decoding for both the hard and soft attention models as it is significantly slower and did not show clear improvement in previous experiments we conducted. For the character level alignment process we use the implementation provided by the organizers of the SIGMORPHON2016 shared task.[5]

LSTM Equations

We used the LSTM variant implemented in the dynet toolkit, which corresponds to the following equations:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{W}_{ic}\mathbf{c}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{f}_{t-1} + \mathbf{W}_{fc}\mathbf{c}_{t-1} + \mathbf{b}_f) \\
\widetilde{\mathbf{c}} &= \tanh(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{c}_t &= \mathbf{c}_{t-1} \circ \mathbf{f}_t + \widetilde{\mathbf{c}} \circ \mathbf{i}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{W}_{ox}\mathbf{c}_t + \mathbf{b}_o) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t) \circ \mathbf{o}_t
\end{aligned}
\tag{6}
$$

[4] https://github.com/clab/dynet

[5] https://github.com/ryancotterell/sigmorphon2016

Improving Sequence to Sequence Learning for Morphological Inflection Generation: The BIU-MIT Systems for the SIGMORPHON 2016 Shared Task for Morphological Reinflection

Roee Aharoni and **Yoav Goldberg**
Computer Science Department
Bar Ilan University
roee.aharoni,yoavgo@gmail.com

Yonatan Belinkov
CSAIL
MIT
belinkov@mit.edu

Abstract

Morphological reinflection is the task of generating a target form given a source form and the morpho-syntactic attributes of the target (and, optionally, of the source). This work presents the submission of Bar Ilan University and the Massachusetts Institute of Technology for the morphological reinflection shared task held at SIGMORPHON 2016. The submission includes two recurrent neural network architectures for learning morphological reinflection from incomplete inflection tables while using several novel ideas for this task: morpho-syntactic attribute embeddings, modeling the concept of templatic morphology, bidirectional input character representations and neural discriminative string transduction. The reported results for the proposed models over the ten languages in the shared task bring this submission to the second/third place (depending on the language) on all three sub-tasks out of eight participating teams, while training only on the Restricted category data.

1 Introduction

Morphological inflection, or reinflection, involves generating a target (surface form) word from a source word (e.g. a lemma), given the morpho-syntactic attributes of the target word. Previous approaches to automatic inflection generation usually make use of manually constructed Finite State Transducers (Koskenniemi, 1983; Kaplan and Kay, 1994), which are theoretically appealing but require expert knowledge, or machine learning methods for string transduction (Yarowsky and Wicentowski, 2000; Dreyer and Eisner, 2011; Durrett and DeNero, 2013; Hulden et al., 2014; Ahlberg et al., 2015; Nicolai et al., 2015). While these studies achieved high accuracies, they also make specific assumptions about the set of possible morphological processes that create the inflection, and require feature engineering over the input.

More recently, Faruqui et al. (2016) used encoder-decoder neural networks for inflection generation inspired by similar approaches for sequence-to-sequence learning for machine translation (Bahdanau et al., 2014; Sutskever et al., 2014). The general idea is to use an encoder-decoder network over characters, that encodes the input lemma into a vector and decodes it one character at a time into the inflected surface word. They factor the data into sets of inflections with identical morpho-syntactic attributes (we refer to each such set as a factor) and try two training approaches: in one they train an individual encoder-decoder RNN per factor, and in the other they train a single encoder RNN over all the lemmas in the dataset and a specific decoder RNN per factor.

An important aspect of previous work on learning inflection generation is the reliance on complete inflection tables – the training data contains all the possible inflections per lemma. In contrast, in the shared task setup (Cotterell et al., 2016) the training is over partial inflection tables that mostly contain only several inflections per lemma, for three different sub-tasks: The first requires morphological inflection generation given a lemma and a set of morpho-syntactic attributes, the second requires morphological re-inflection of an inflected word given the word, its morpho-syntactic attributes and the target inflection's attributes, and the third requires re-inflection of an inflected word given only the target inflection attributes. The datasets for the different tasks are available on the

Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 41–48,
Berlin, Germany, August 11, 2016. ©2016 Association for Computational Linguistics

shared task's website.[1]

The fact that the data is incomplete makes it problematic to use factored models like the ones introduced in (Faruqui et al., 2016), as there may be insufficient data for training a high-quality model per factor of inflections with identical morpho-syntactic attributes. For example, in the shared task dataset the training data usually contains less than 100 training examples on average per such factor. Moreover, when the data is factored this way, no information is shared between the different factors even though they may have identical inflection rules.

We propose two neural network architectures for the task. The first, detailed in Section 2, departs from the architecture of (Faruqui et al., 2016) by extending it in three novel ways: representing morpho-syntactic attributes, template-inspired modeling, and bidirectional input character representations. The second, described in Section 3, is based on an explicit control mechanism we introduce while also making use of the three extensions mentioned above. Our experimental evaluation over all 10 languages represented in the shared task brings our models to the second or third place, depending on the language.

2 First Approach: Morphological Sequence to Sequence Architecture

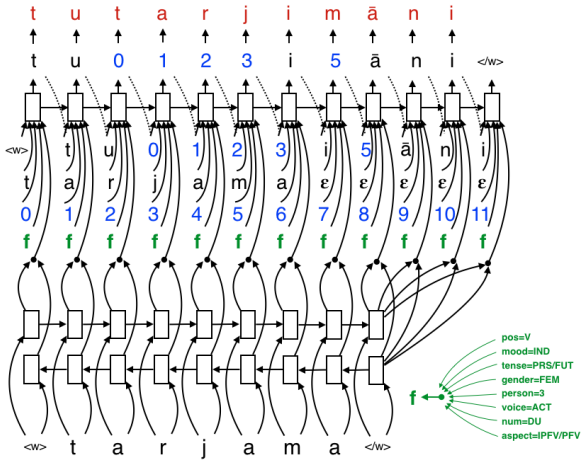
Our first proposed architecture is a Morphological Sequence to Sequence (MS2S) architecture, illustrated in Figure 1. It incorporates several novel components in the sequence-to-sequence learning paradigm, as discussed below.

2.1 Morpho-Syntactic Attribute Embeddings

We seek to train models over larger amounts of examples, rather than on factors that strictly contain examples that share all the morpho-syntactic attributes. To do so, instead of factoring the data by the attributes we feed the attributes into the network by creating a dense embedding vector for every possible attribute/value pair (for example, gender=FEM and gender=MASC will each have its own embedding vector). The attribute embeddings for each input are then concatenated and added as parameters to the network while being updated during training similarly to the character embeddings. This way, information can be shared

[1] http://ryancotterell.github.io/sigmorphon2016/

Figure 1: The Morphological Sequence to Sequence (MS2S) network architecture for predicting an inflection template given the Arabic lemma *tarjama* and a set of morpho-syntactic attributes. A round tip expresses concatenation of the inputs it receives.



across inflections with different morpho-syntactic attributes, as they are trained jointly, while the attribute embeddings help discriminate between different inflection types when needed. This can be seen in Figure 1, where f is the vector containing a concatenation of the morpho-syntactic attribute embeddings.

While this approach should allow us to train a single neural network over the entire dataset to predict all the different inflection types, in practice we were not able to successfully train such a network. Instead, we found a middle ground in training a network per part-of-speech (POS) type. This resulted in much fewer models than in the factored model, each using much more data, which is essential when training machine learning models and specifically neural networks. For example, on the Arabic dataset of the first sub task (inflection generation from lemma to word) this reduced the amount of trained models from 223 with an average of 91 training examples per model, to only 3 models (one per POS type - verb, noun, adjective) with an average of 3907 training examples per model.

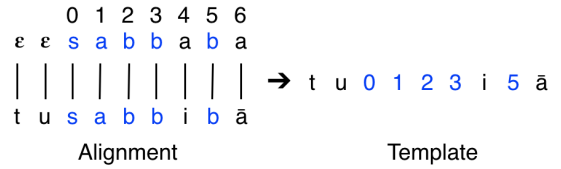
2.2 Morphological Templates

We bring the idea of morphological templates into the model: instead of training the network to predict only a specific inflection character at each step given a lemma and a set of morpho-syntactic features, we train the network to either predict a char-

acter from the vocabulary or *to copy* a character at a given position in the input sequence. This enables the network to produce a sequence that resembles a morphological template which can be instantiated with characters from the input to produce the correct inflection. While at train time we encourage the network to perform copy operations when possible, at prediction time the network can decide whether to copy a character from the input by predicting its location in the input or to generate a preferred character from the vocabulary. For example, for the Arabic lemma *tarjama* and a set of morpho-syntactic attributes the network will output the sequence "tu0123i5āni" which can be instantiated with the lemma into the correct inflection, *tutarjimāni*, as depicted in Figure 1.

Intuitively, this method enables the learning process to generalize better as many different examples may share similar templates – which is important when working with relatively small datasets. We saw indeed that adding this component to our implementation of a factored model similar to (Faruqui et al., 2016) gave a significant improvement in accuracy over the Arabic dataset: from 24.04 to 78.35, while the average number of examples per factor was 91.

To implement this, for every given pair of input and output sequences in the training set we need to produce a parameterized sequence which, when instantiated with the input sequence, creates the output sequence. This is achieved by running a character level alignment process on the training data, which enables to easily infer the desired sequence from every input-output sequence alignment. For example, given the input sequences *sabbaba* and output sequence *tusabbibā* with the induced alignment $\epsilon\epsilon$sabbaba-tusabbibā, we produce the expected output: tu0123i5ā, as depicted in the next figure:



We performed the alignment process using a Chinese Restaurant Process character level aligner (Sudoh et al., 2013) as implemented in the shared task baseline system.[2]

[2] https://github.com/ryancotterell/sigmorphon2016/tree/master/src/baseline

2.3 Bidirectional Input Character Representation

Instead of feeding the decoder RNN at each step with a fixed vector that holds the encoded vector for the entire input sequence like Faruqui et. al. (2016), we feed the decoder RNN at each step with a Bi-Directional Long-Short Term Memory (BiLSTM) representation (Graves and Schmidhuber, 2005) per character in the input along with the character embedding learned by the network. The BiLSTM character representation is a concatenation of the outputs of two LSTMs that run over the character sequence up to the current character, from both sides. This adds more focused context when the network predicts the next inflection output, while still including information form the entire sequence due to the bidirectional representation.

2.4 MS2S Decoder Input

For every step i of the decoder RNN for this setup, the input vector is a concatenation of the following:

1. $BiLSTM_i$ – The bidirectional character embedding for the ith input character (if i is larger than the length of the input sequence, the embedding of the last input character is used).

2. c_i – The character embedding for the ith input character. If i is larger than the length of the input sequence, an embedding of a special $\mathcal{E}$ symbol is used, similarly to (Faruqui et al., 2016).

3. i – A character embedding for the current step index in the decoder. In the first step this will be an embedding matching to '0', in the second step it will an embedding matching to '1' etc. These are the same index embeddings used to model copy actions from a specific index.

4. o_{i-1} – The feedback input, containing the embedding of the prediction (either a character or an integer representing an index in the input) from the previous decoder RNN step.

5. f – The vector containing the concatenation of the morpho-syntactic attribute embeddings.

3 Second Approach: The Neural Discriminative String Transducer Architecture

The second approach is based on a Neural Discriminative String Transducer (NDST), a novel neural network architecture that models which specific part of the input sequence is relevant for predicting the next output character at a given time. This is done by maintaining a state consisting of an input sequence position (the input pointer) and an output sequence position (the output pointer), which are controlled by the decoder. This approach can be seen as a more focused replacement to the general attention mechanism of Bahdanau et. al. (2014), tailored to the usually monotonic behavior of the output sequence with respect to the input sequence in the morphological reinflection task. An example for using this architecture is available in Figure 2.

3.1 NDST Decoder Input

For every step i in the NDST decoder RNN, the input vector is a concatenation of the following:

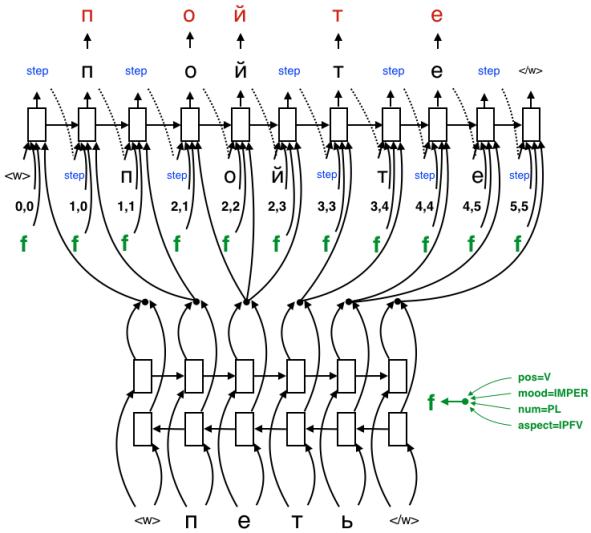
1. p_{input} – The input pointer, holding the embedding that represents the position of the current pointed input sequence element. When $i = 0$, this is initialized with the embedding that stands for the position of the first element in the input. Every time the network outputs the "step" symbol, p_{input} is promoted by setting it with the embedding that represents the next input sequence position.

2. p_{output} – The output pointer, a character embedding representing the next position in the output sequence to be generated. When $i = 0$, this is initialized with the embedding that stands for the position of the first element in the input. Every time the network outputs a symbol other than the "step" symbol, p_{output} is promoted by setting it with the embedding for the next output sequence position.

3. $BiLSTM_{p_{input}}$ – The bidirectional character embedding for the input character currently pointed by p_{input}.

4. o_{i-1} – The feedback input, containing the embedding of the prediction (either a character, an integer representing an index in the input, or the "step" symbol) from the previous decoder RNN step.

Figure 2: The Neural Discriminative String Transducer (NDST) architecture for predicting an inflection template given a lemma and a set of morpho-syntactic attributes.



5. f – The vector containing the concatenation of the morpho-syntactic attribute embeddings.

To train an NDST network, for every input and output sequence in the training data we should have a sequence of actions (of three types – either a specific character prediction, an index to copy from or a "step" instruction) that when performed on the input sequence, produces the correct output sequence. To get the correct instruction sequences in train time we first run a character level alignment process on the training data, similarly to the MS2S model. Once we have the character level alignment per input-output sequence pair, we deterministically infer the sequence of actions that results in the desired output by going through every pair of aligned input-output characters in the alignment. If the input and output characters in the aligned pair are not identical, we produce the new output character. If the input and output characters in the aligned pair are identical we produce a copy action from the input character location. After that, if the next output character is not the epsilon symbol as seen in the alignment in Figure 2.2 we also produce a "step" action. We train the network to produce this sequence of actions when given the input sequence and the set of morpho-syntactic attributes matching the desired inflection.

4 Experimental Details

4.1 Submissions

The shared task allowed submissions in three different tracks: Standard, which enabled using data from lower numbered tasks in addition to the current task data; Restricted, which enabled using only the current task's data; and Bonus, which enabled using the Standard track datasets and an additional monolingual corpus supplied by the organizers.

We submitted two systems to the shared task, both in the Restricted track: The first, named BIU/MIT-1, used the MS2S architecture as described previously and participated in all three sub-tasks. Notice that for the 3rd task, the input is identical to the first task so it does not require changes in the network architecture. To use the MS2S network for the second task we concatenated the source and target morpho-syntactic attribute embeddings and used that vector as the f vector mentioned previously. The output from this system was 5-best lists, meaning 5 predictions for each input. To produce the 5-best list we perform beam search over the MS2S model, which is trained greedily without such search procedure.

The second system, named BIU/MIT-2, used the NDST architecture and participated only in the first and second sub-tasks. This system did not use beam search, producing only one guess per input. Again, to use the NDST architecture for the second task we simply concatenated the input and output morpho-syntactic attribute embeddings.

4.2 Training, Implementation and Hyper Parameters

To train our systems, we used the train portion of the dataset as-is and submitted the model which performed best on the development portion of the dataset, without conducting any specific preprocessing steps on the data. We trained our networks for a maximum of 300 epochs over the entire training set or until no improvement on the development set has been observed for more than 100 epochs. The systems were implemented using pyCNN, the python wrapper for the CNN toolkit.[3] In both architectures we trained the network by optimizing the expected output sequence likelihood using cross-entropy loss. For optimization we used ADAM (Kingma and Ba, 2014) with no regularization, and the parameters set as $\alpha = 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. In all architectures we used the CNN toolkit implementation of an LSTM network with two layers, each having 200 entries. The character embeddings were also vectors with 200 entries, and the morpho-syntactic attribute embeddings were vectors of 20 entries. When using beam search we used a beam width of 5.

[3] https://github.com/clab/cnn

5 Results

While developing our systems we measured our performance on the development set with respect to two baselines: the shared task baseline system (ST-Base) inspired by (Nicolai et al., 2015; Durrett and DeNero, 2013), and the factored sequence to sequence baseline (Fact.) similar to the one introduced in (Faruqui et al., 2016). On the test set, our systems ranked second or third out of eight groups in the shared task (depending on the language). The best participating system, LMU-1/2 (Kann and Schütze, 2016) relied on a single encoder-decoder model with attention (Bahdanau et al., 2014) per language, with several improvements like performing prediction using majority voting over an ensemble of five models. In contrast, our first system did not use an explicit attention mechanism and is composed of 3 models per language (one per POS type) without using ensembling. We compare our system to the best system on the test set.

The results for the first task are shown in Table 1, measuring aggregated accuracy across all POS tags. On the development set, our models surpassed both baselines significantly and were competitive with each other, as the MS2S model gained the best aggregated accuracy results on all languages but Russian and Finnish, where the NDST model was better. On the test set, similar results are shown: the MS2S model gives higher accuracies except for Russian, Navajo and Maltese where the NDST model was superior.

For the second task, we measured performance only with respect to ST-Base as can be seen in Table 2. On the development set, the NDST model outperformed the baseline and the MS2S model for all languages but Georgian and Spanish, where the MS2S and ST-Base models were better, respectively, although not with a significant difference. On the test set, the MS2S model gave better results only for Georgian and Hungarian.

Table 1: Results for inflection generation (first sub-task), measuring accuracy on the development set: our models vs. the shared task (ST-Base) and Factored (Fact.) baselines, and mean reciprocal rank (MRR) on the test set: our models vs. the best performing model (Kann and Schütze, 2016).

	Dev				Test		
Language	ST-Base	Fact.	MS2S	NDST	MS2S	NDST	Best
Russian	90.38	84.22	91.57	**93.33**	89.73	90.62	**91.46**
Georgian	89.83	92.37	**98.41**	97.01	97.55	96.54	**98.5**
Finnish	68.27	75.78	**95.8**	94.36	93.81	92.58	**96.8**
Arabic	70.29	24.04	**96.28**	92.95	93.34	89.96	**95.47**
Navajo	71.9	83.47	**98.82**	98.48	80.13	88.43	**91.48**
Spanish	96.92	91.79	98.99	**99.31**	98.41	98.33	**98.84**
Turkish	59.17	64.68	**98.18**	97.8	97.74	96.17	**98.93**
German	89.29	90.35	**96.36**	95.99	95.11	94.87	**95.8**
Hungarian	78.62	65.75	**99.23**	98.76	98.33	97.59	**99.3**
Maltese	36.94	N/A	**87.92**	85.2	82.4	84.78	**88.99**

Table 2: Results for morphological re-inflection with source attributes (second sub-task) measuring accuracy over the development set: our models vs. the shared task (ST-Base) baseline, and mean reciprocal rank (MRR) over the test set: our models vs. the best performing model (Kann and Schütze, 2016)

	Dev			Test		
Language	ST-Base	MS2S	NDST	MS2S	NDST	Best
Russian	85.63	85.06	**86.62**	83.36	85.81	**90.11**
Georgian	91.5	**94.13**	93.81	92.65	92.27	**98.5**
Finnish	64.56	77.13	**84.31**	74.44	80.91	**96.81**
Arabic	58.75	75.25	**78.37**	70.26	73.95	**91.09**
Navajo	60.85	63.85	**75.04**	56.5	67.88	**97.81**
Spanish	**95.63**	93.25	95.37	92.21	94.26	**98.45**
Turkish	54.88	82.56	**87.25**	81.69	83.88	**98.38**
German	87.69	93.13	**94.12**	91.67	92.66	**96.22**
Hungarian	78.33	94.37	**94.87**	92.33	91.16	**99.42**
Maltese	26.2	43.29	**49.7**	41.92	50.13	**86.88**

Table 3: Results for morphological re-inflection without source attributes (third sub-task) measuring accuracy over the development set: our models vs. the shared task (ST-Base) baseline, and mean reciprocal rank (MRR) over the test set: our models vs. the best performing model (Kann and Schütze, 2016)

	Dev			Test	
Language	ST-Base	MS2S	NDST	MS2S	Best
Russian	81.31	**84.56**	84.25	82.81	**87.13**
Georgian	90.68	**93.62**	91.05	92.08	**96.21**
Finnish	61.94	**76.5**	66.25	72.99	**93.18**
Arabic	50	**72.56**	69.31	69.05	**82.8**
Navajo	60.26	**62.7**	54.0	52.85	**83.5**
Spanish	88.94	**92.62**	89.68	92.14	**96.69**
Turkish	52.19	**79.87**	75.25	79.69	**95.0**
German	81.56	**90.93**	89.31	89.58	**92.41**
Hungarian	78	**94.25**	83.83	91.91	**98.37**
Maltese	24.75	**44.04**	3.58	40.79	**84.25**

For the third task we also measured performance with respect to ST-Base as can be seen in Table 3. On the development set, the MS2S model outperformed the others on all languages. Since this was the situation we did not submit the NDST model for this sub-task, thus not showing test results for the NDST model on the test set.

6 Preliminary Analysis

An obvious trend we can see in the results is the MS2S approach giving higher accuracy scores on the first and third tasks, while the NDST approach being significantly better on the second task. While inspecting the data for the second and third tasks we noticed that the datasets only differ in the added morpho-syntactic attributes for the input sequences, and are identical other then that. This is encouraging as it shows how the NDST control mechanism can facilitate the additional data on the input sequence to predict inflections in a better way. We plan to further analyze the results to better understand the cases where the NDST architecture provides added value over the MS2S approach.

7 Discussion and Future Work

Our systems reached the second/third place in the Restricted category in the shared task, depending on the language/sub-task combination. It is also encouraging to see that if we submitted our systems as-is to the Standard and Bonus tracks we would also get similar rankings, even without using the additional training data available there. The winning submission in all tracks, described in (Kann and Schütze, 2016) also used an encoder-decoder approach that incorporated the morpho-syntactic attributes as inputs to the network, but with several differences from our approach like using an attention mechanism similar to (Bahdanau et al., 2014), training a single model for all inflection types rather than one per POS type and performing prediction by using an ensemble of five models with majority voting rather than using a single trained model like we did. Future work may include exploring a hybrid approach that combines the ideas proposed in our work and the latter. Other recent works that propose ideas relevant to explore in future work in this direction are (Gu et al., 2016), which describe a different copying mechanism for encoder-decoder architectures, or (Rastogi et al., 2016), which models the reinflection task using a finite state transducer weighted with neural context that also takes special care of the character copying issue.

Acknowledgments

We thank Manaal Faruqui for sharing his code with us. This work was supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *HLT-NAACL*, pages 1024–1029. The Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany, August. The Association for Computational Linguistics.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *EMNLP*, pages 616–627. The Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia, June. The Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, USA, June 12 - June 17, 2016*.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Mans Hulden, Markus Forsberg, and Malin Ahlberg. 2014. Semi-supervised learning of morphological paradigms and lexicons. In Gosse Bouma and Yannick Parmentier 0001, editors, *EACL*, pages 569–578. The Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *ACL*, Berlin, Germany, August. The Association for Computational Linguistics.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Kimmo Koskenniemi. 1983. Two-level morphology: A general computational model of word-form recognition and production. Technical Report Publication No. 11, Department of General Linguistics, University of Helsinki.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado, May–June. The Association for Computational Linguistics.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proc. of NAACL*.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2013. Noise-aware character alignment for bootstrapping statistical machine transliteration from bilingual corpora. In *EMNLP*, pages 204–209. The Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *NIPS*, pages 3104–3112.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *ACL*. The Association for Computational Linguistics.

Chapter 4

Linearizing Syntax for String-to-Tree Neural Machine Translation

Towards String-to-Tree Neural Machine Translation

Roee Aharoni & Yoav Goldberg
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{roee.aharoni,yoav.goldberg}@gmail.com

Abstract

We present a simple method to incorporate syntactic information about the target language in a neural machine translation system by translating into linearized, lexicalized constituency trees. Experiments on the WMT16 German-English news translation task shown improved BLEU scores when compared to a syntax-agnostic NMT baseline trained on the same dataset. An analysis of the translations from the syntax-aware system shows that it performs more reordering during translation in comparison to the baseline. A small-scale human evaluation also showed an advantage to the syntax-aware system.

1 Introduction and Model

Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014) has recently became the state-of-the-art approach to machine translation (Bojar et al., 2016), while being much simpler than the previously dominant phrase-based statistical machine translation (SMT) approaches (Koehn, 2010). NMT models usually do not make explicit use of syntactic information about the languages at hand. However, a large body of work was dedicated to syntax-based SMT (Williams et al., 2016). One prominent approach to syntax-based SMT is string-to-tree (S2T) translation (Yamada and Knight, 2001, 2002), in which a source-language string is translated into a target-language tree. S2T approaches to SMT help to ensure the resulting translations have valid syntactic structure, while also mediating flexible reordering between the source and target languages. The main formalism driving current S2T SMT systems is GHKM rules (Galley et al., 2004, 2006), which are synchronous transduction grammar (STSG) fragments, extracted from word-aligned sentence pairs with syntactic trees on one side. The GHKM translation rules allow flexible reordering on all levels of the parse-tree.

We suggest that NMT can also benefit from the incorporation of syntactic knowledge, and propose a simple method of performing string-to-tree neural machine translation. Our method is inspired by recent works in syntactic parsing, which model trees as sequences (Vinyals et al., 2015; Choe and Charniak, 2016). Namely, we translate a source sentence into a linearized, lexicalized constituency tree, as demonstrated in Figure 2. Figure 1 shows a translation from our neural S2T model compared to one from a vanilla NMT model for the same source sentence, as well as the attention-induced word alignments of the two models.

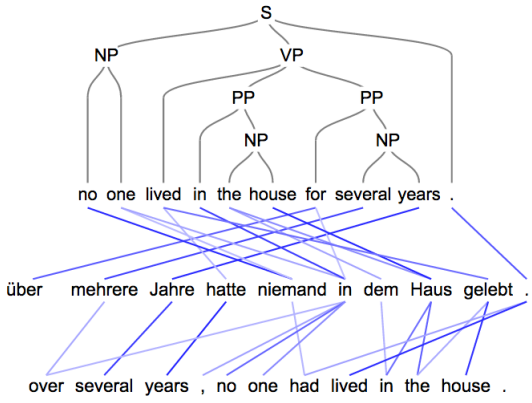


Figure 1: Top - a lexicalized tree translation predicted by the bpe2tree model. Bottom - a translation for the same sentence from the bpe2bpe model. The blue lines are drawn according to the attention weights predicted by each model.

Note that the linearized trees we predict are different in their structure from those in Vinyals et al. (2015) as instead of having part of speech tags as terminals, they contain the words of the translated sentence. We intentionally omit the POS informa-

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 132–140
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2021

Jane hatte eine Katze . $\rightarrow$ $(_{ROOT}$ $(_{S}$ $(_{NP}$ **Jane** $)_{NP}$ $(_{VP}$ **had** $(_{NP}$ **a cat** $)_{NP}$ $)_{VP}$ **.** $)_{S}$ $)_{ROOT}$

Figure 2: An example of a translation from a string to a linearized, lexicalized constituency tree.

tion as including it would result in significantly longer sequences. The S2T model is trained on parallel corpora in which the target sentences are automatically parsed. Since this modeling keeps the form of a sequence-to-sequence learning task, we can employ the conventional attention-based sequence to sequence paradigm (Bahdanau et al., 2014) as-is, while enriching the output with syntactic information.

Related Work Some recent works did propose to incorporate syntactic or other linguistic knowledge into NMT systems, although mainly on the source side: Eriguchi et al. (2016a,b) replace the encoder in an attention-based model with a Tree-LSTM (Tai et al., 2015) over a constituency parse tree; Bastings et al. (2017) encoded sentences using graph-convolutional networks over dependency trees; Sennrich and Haddow (2016) proposed a factored NMT approach, where each source word embedding is concatenated to embeddings of linguistic features of the word; Luong et al. (2015) incorporated syntactic knowledge via multi-task sequence to sequence learning: their system included a single encoder with multiple decoders, one of which attempts to predict the parse-tree of the source sentence; Stahlberg et al. (2016) proposed a hybrid approach in which translations are scored by combining scores from an NMT system with scores from a Hiero (Chiang, 2005, 2007) system. Shi et al. (2016) explored the syntactic knowledge encoded by an NMT encoder, showing the encoded vector can be used to predict syntactic information like constituency trees, voice and tense with high accuracy.

In parallel and highly related to our work, Eriguchi et al. (2017) proposed to model the target syntax in NMT in the form of dependency trees by using an RNNG-based decoder (Dyer et al., 2016), while Nadejde et al. (2017) incorporated target syntax by predicting CCG tags serialized into the target translation. Our work differs from those by modeling syntax using constituency trees, as was previously common in the "traditional" syntax-based machine translation literature.

2 Experiments & Results

Experimental Setup We first experiment in a resource-rich setting by using the German-English portion of the WMT16 news translation task (Bojar et al., 2016), with 4.5 million sentence pairs. We then experiment in a low-resource scenario using the German, Russian and Czech to English training data from the News Commentary v8 corpus, following Eriguchi et al. (2017). In all cases we parse the English sentences into constituency trees using the BLLIP parser (Charniak and Johnson, 2005).[1] To enable an open vocabulary translation we used sub-word units obtained via BPE (Sennrich et al., 2016b) on both source and target.[2]

In each experiment we train two models. A **baseline** model (bpe2bpe), trained to translate from the source language sentences to English sentences without any syntactic annotation, and a **string-to-linearized-tree** model (bpe2tree), trained to translate into English linearized constituency trees as shown in Figure 2. Words are segmented into sub-word units using the BPE model we learn on the raw parallel data. We use the NEMATUS (Sennrich et al., 2017)[3] implementation of an attention-based NMT model.[4] We trained the models until there was no improvement on the development set in 10 consecutive checkpoints. Note that the only difference between the baseline and the bpe2tree model is the syntactic information, as they have a nearly-identical amount of model parameters (the only additional parameters to the syntax-aware system are the embeddings for the brackets of the trees).

For all models we report results of the best performing single model on the dev-set (newstest2013+newstest2014 in the resource rich setting, newstest2015 in the rest, as measured by BLEU) when translating newstest2015 and newstest2016, similarly to Sennrich et al. (2016a); Eriguchi et al. (2017). To evaluate the string-to-tree translations we derive the surface form by removing the symbols that stand for non-terminals in the tree, followed by merging the sub-words. We also report the results of an ensemble of the last 5 checkpoints saved during each model training. We compute BLEU scores using the

[1] https://github.com/BLLIP/bllip-parser

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

[3] https://github.com/rsennrich/nematus

[4] Further technical details of the setup and training are available in the supplementary material.

`mteval-v13a.pl` script from the Moses toolkit (Koehn et al., 2007).

system	newstest2015	newstest2016
bpe2bpe	27.33	31.19
bpe2tree	27.36	32.13
bpe2bpe ens.	28.62	32.38
bpe2tree ens.	28.7	33.24

Table 1: BLEU results for the WMT16 experiment

Results As shown in Table 1, for the resource-rich setting, the single models (bpe2bpe, bpe2tree) perform similarly in terms of BLEU on newstest2015. On newstest2016 we witness an advantage to the bpe2tree model. A similar trend is found when evaluating the model ensembles: while they improve results for both models, we again see an advantage to the bpe2tree model on newstest2016. Table 2 shows the results in the low-resource setting, where the bpe2tree model is consistently better than the bpe2bpe baseline. We find this interesting as the syntax-aware system performs a much harder task (predicting trees on top of the translations, thus handling much longer output sequences) while having a nearly-identical amount of model parameters. In order to better understand where or how the syntactic information improves translation quality, we perform a closer analysis of the WMT16 experiment.

3 Analysis

The Resulting Trees Our model produced valid trees for 5970 out of 6003 sentences in the development set. While we did not perform an in-depth error-analysis, the trees seem to follow the syntax of English, and most choices seem reasonable.

Quantifying Reordering English and German differ in word order, requiring a significant amount of reordering to generate a fluent translation. A major benefit of S2T models in SMT is facilitating reordering. Does this also hold for our neural S2T model? We compare the amount of reordering in the bpe2bpe and bpe2tree models using a distortion score based on the alignments derived from the attention weights of the corresponding systems. We first convert the attention weights to hard alignments by taking for each target word the source word with highest attention weight. For an n-word target sentence t and source sentence s let $a(i)$ be the position of the source word aligned to the target word in position i. We define:

	system	newstest2015	newstest2016
DE-EN	bpe2bpe	13.81	14.16
	bpe2tree	14.55	16.13
	bpe2bpe ens.	14.42	15.07
	bpe2tree ens.	15.69	17.21
RU-EN	bpe2bpe	12.58	11.37
	bpe2tree	12.92	11.94
	bpe2bpe ens.	13.36	11.91
	bpe2tree ens.	13.66	12.89
CS-EN	bpe2bpe	10.85	11.23
	bpe2tree	11.54	11.65
	bpe2bpe ens.	11.46	11.77
	bpe2tree ens.	12.43	12.68

Table 2: BLEU results for the low-resource experiments (News Commentary v8)

$$d(s,t) = \frac{1}{n}\sum_{i=2}^{n} |a(i) - a(i-1)|$$

For example, for the translations in Figure 1, the above score for the bpe2tree model is 2.73, while the score for the bpe2bpe model is 1.27 as the bpe2tree model did more reordering. Note that for the bpe2tree model we compute the score only on tokens which correspond to terminals (words or sub-words) in the tree. We compute this score for each source-target pair on newstest2015 for each model. Figure 3 shows a histogram of the binned score counts. The bpe2tree model has more translations with distortion scores in bins 1-onward and significantly less translations in the least-reordering bin (0) when compared to the bpe2bpe model, indicating that the syntactic information encouraged the model to perform more reordering.[5] Figure 4 tracks the distortion scores throughout the learning process, plotting the average dev-set scores for the model checkpoints saved every 30k updates. Interestingly, both models obey to the following trend: open with a relatively high distortion score, followed by a steep decrease, and from there ascend gradually. The bpe2tree model usually has a higher distortion score during training, as we would expect after our previous findings from Figure 3.

Tying Reordering and Syntax The bpe2tree model generates translations with their constituency tree and their attention-derived alignments. We can use this information to extract GHKM rules (Galley et al., 2004).[6] We derive

[5] We also note that in bins 4-6 the bpe2bpe model had slightly more translations, but this was not consistent among different runs, unlike the gaps in bins 0-3 which were consistent and contain most of the translations.

[6] `github.com/joshua-decoder/galley-ghkm`

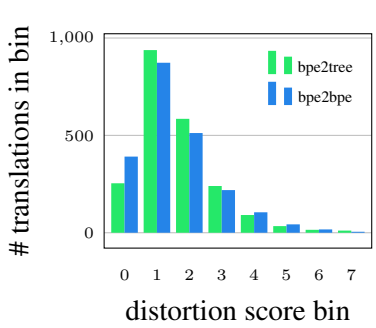


Figure 3: newstest2015 DE-EN translations binned by distortion amount

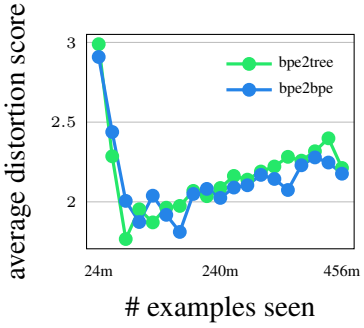


Figure 4: Average distortion score on the dev-set during different training stages

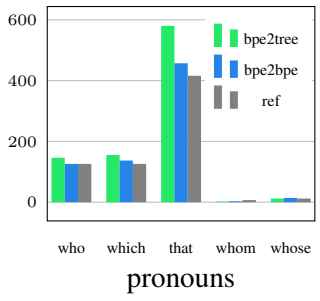


Figure 5: Amount of English relative pronouns in newstest2015 translations

LHS	**Top-5 RHS, sorted according to count.**
VP(x0:TER x1:NP)	(244) x0 x1 (157) **x1 x0** (80) x0 x1 ”,/.” (56) **x1 x0 ”,/.”** (17) x0 ”eine” x1
VP(x0:TER PP(x1:TER x2:NP))	(90) **x1 x2 x0** (65) x0 x1 x2 (31) **x1 x2 x0 ”,/.”** (13) x0 x1 x2 ”,/.” (7) **x1 ”der” x2 x0**
VP(x0:TER x1:PP)	(113) **x1 x0** (82) x0 x1 (38) **x1 x0 ”,/.”** (18) x0 x1 ”,/.” (5) ”,/.” x0 x1
S(x0:NP VP(x1:TER x2:NP))	(69) x0 x1 x2 (51) **x0 x2 x1** (35) x0 x1 x2 ”,/.” (20) **x0 x2 x1 ”,/.”** (6) ”die” x0 x1 x2
VP(x0:TER x1:NP x2:PP)	(52) x0 x1 x2 (38) **x1 x2 x0** (20) **x1 x2 x0 ”,/.”** (11) x0 x1 x2 ”,/.” (9) **x2 x1 x0**
VP(x0:TER x1:NP PP(x2:TER x3:NP))	(40) x0 x1 x2 x3 (32) **x1 x2 x3 x0** (18) **x1 x2 x3 x0 ”,/.”** (8) x0 x1 x2 x3 ”,/.” (5) **x2 x3 x1 x0**
VP(x0:TER NP(x1:NP x2:PP))	(61) x0 x1 x2 (38) **x1 x2 x0** (19) x0 x1 x2 ”,/.” (8) x0 ”eine” x1 x2 (8) **x1 x2 x0 ”,/.”**
NP(x0:NP PP(x1:TER x2:NP))	(728) x0 x1 x2 (110) ”die” x0 x1 x2 (107) x0 x1 x2 ”,/.” (56) x0 x1 ”der” x2 (54) ”der” x0 x1 x2
S(VP(x0:TER x1:NP))	(41) **x1 x0** (26) x0 x1 (14) **x1 x0 ”,/.”** (7) **”die” x1 x0** (5) x0 x1 ”,/.”
VP(x0:TER x1:VP)	(73) x0 x1 (38) **x1 x0** (25) x0 x1 ”,/.” (15) **x1 x0 ”,/.”** (9) ”,/.” x0 x1

Table 3: Top dev-set GHKM Rules with reordering. Numbers: rule counts. Bolded: reordering rules.

src	Dutzende türkischer Polizisten wegen ”Verschwörung” gegen die Regierung festgenommen	
ref	Tens of Turkish Policemen Arrested over ’Plotting’ against Gov’t	
2tree	dozens of Turkish police arrested for ”conspiracy” against the government.	
2bpe	dozens of turkish policemen on ”conspiracy” against the government arrested	
src	Die Menschen in London weinten, als ich unsere Geschichte erzhlte.	Er ging einen Monat nicht zu Arbeit.
ref	People in London were crying when I told our story.	He ended up spending a month off work.
2tree	the people of london wept as I told our story.	he did not go to work a month.
2bpe	the people of London, when I told our story.	he went one month to work.
src	Achenbach habe für 121 Millionen Euro Wertgegenstände für Albrecht angekauft.	
ref	Achenbach purchased valuables for Albrecht for 121 million euros.	
2tree	Achenbach has bought valuables for Albrecht for 121 million euros.	
2bpe	Achenbach have purchased value of 121 million Euros for Albrecht.	
src	Apollo investierte 2008 1 Milliarde $ in Norwegian Cruise.	Könntest du mal mit dem ”ich liebe dich” aufhören?
ref	Apollo made a $1 billion investment in Norwegian Cruise in 2008.	Could you stop with the ”I love you”?
2tree	Apollo invested EUR $1 billion in Norwegian Cruise in 2008.	Could you stop saying ”I love you?
2bpe	Apollo invested 2008 $1 billion in Norwegian Cruise.	Can you say with the ”I love you” stop?
src	Gerade in dieser schweren Phase hat er gezeigt, dass er für uns ein sehr wichtiger Spieler ist”, konstatierte Barisic.	
ref	Especially during these difficult times, he showed that he is a very important player for us”, Barisic stated.	
2tree	Especially at this difficult time he has shown that he is a very important player for us,” said Barisic.	
2bpe	It is precisely during this difficult period that he has shown us to be a very important player, ”Barisic said.	
src	Hopfen und Malz - auch in China eine beliebte Kombination.	”Ich weiß jetzt, dass ich das kann - prima!”
ref	Hops and malt - a popular combination even in China.	”I now know that I can do it - brilliant!”
2tree	Hops and malt - a popular combination in China.	”I now know that I can do that!
2bpe	Hops and malt - even in China, a popular combination.	I know now that I can that - prima!”
src	Die Ukraine hatte gewarnt, Russland könnte auch die Gasversorgung für Europa unterbrechen.	
ref	Ukraine warned that Russia could also suspend the gas supply to Europe.	
2tree	Ukraine had warned that Russia could also stop the supply of gas to Europe.	
2bpe	Ukraine had been warned, and Russia could also cut gas supplies to Europe.	
src	Bis dahin gab es in Kollbach im Schulverband Petershausen-Kollbach drei Klassen und in Petershausen fünf.	
ref	Until then, the school district association of Petershausen-Kollbach had three classes in Kollbach and five in Petershausen.	
2tree	until then, in Kollbach there were three classes and five classes in Petershausen.	
2bpe	until then there were three classes and in Petershausen five at the school board in Petershausen-Kollbach.	

Table 4: Translation examples from newstest2015. The underlines correspond to the source word attended by the first opening bracket (these are consistently the main verbs or structural markers) and the target words this source word was most strongly aligned to. See the supplementary material for an attention weight matrix example when predicting a tree (Figure 6) and additional output examples.

hard alignments for that purpose by treating every source/target token-pair with attention score above 0.5 as an alignment. Extracting rules from the dev-set predictions resulted in 233,657 rules, where 22,914 of them (9.8%) included reordering, i.e. contained variables ordered differently in the source and the target. We grouped the rules by their LHS (corresponding to a target syntactic structure), and sorted them by the total number of RHS (corresponding to a source sequential structure) with reordering. Table 3 shows the top 10 extracted LHS, together with the top-5 RHS, for each rule. The most common rule, VP(x_0:TER x_1:NP) $\rightarrow$ x_1 x_0, found in 184 sentences in the dev set (8.4%), is indicating that the sequence x_1 x_0 in German was reordered to form a verb phrase in English, in which x_0 is a terminal and x_1 is a noun phrase. The extracted GHKM rules reveal very sensible German-English reordering patterns.

Relative Constructions Browsing the produced trees hints at a tendency of the syntax-aware model to favor using relative-clause structures and subordination over other syntactic constructions (i.e., "several cameras *that* are all priced..." vs. "several cameras, all priced..."). To quantify this, we count the English relative pronouns (who, which, that[7], whom, whose) found in the newstest2015 translations of each model and in the reference translations, as shown in Figure 5. The bpe2tree model produces more relative constructions compared to the bpe2bpe model, and both models produce more such constructions than found in the reference.

Main Verbs While not discussed until this point, the generated opening and closing brackets also have attention weights, providing another opportunity to to peak into the model's behavior. Figure 6 in the supplementary material presents an example of a complete attention matrix, including the syntactic brackets. While making full sense of the attention patterns of the syntactic elements remains a challenge, one clear trend is that opening the very first bracket of the sentence *consistently attends to the main verb or to structural markers* (i.e. question marks, hyphens) in the source sentence, suggesting a planning-ahead behavior of the decoder. The underlines in Table 4 correspond to the source word attended by the first opening bracket, and the target word this source word was most strongly aligned to. In general, we find the alignments from the syntax-based system more sensible (i.e. in Figure 1 – the bpe2bpe alignments are off-by-1).

Qualitative Analysis and Human Evaluations The bpe2tree translations read better than their bpe2bpe counterparts, both syntactically and semantically, and we highlight some examples which demonstrate this. Table 4 lists some representative examples, highlighting improvements that correspond to syntactic phenomena involving reordering or global structure. We also performed a small-scale human-evaluation using mechanical turk on the first 500 sentences in the dev-set. Further details are available in the supplementary material. The results are summarized in the following table:

2bpe weakly better	100
2bpe strongly better	54
2tree weakly better	122
2tree strongly better	64
both good	26
both bad	3
disagree	131

As can be seen, in 186 cases (37.2%) the human evaluators preferred the bpe2tree translations, vs. 154 cases (30.8%) for bpe2bpe, with the rest of the cases (30%) being neutral.

4 Conclusions and Future Work

We present a simple string-to-tree neural translation model, and show it produces results which are better than those of a neural string-to-string model. While this work shows syntactic information about the target side can be beneficial for NMT, this paper only scratches the surface with what can be done on the subject. First, better models can be proposed to alleviate the long sequence problem in the linearized approach or allow a more natural tree decoding scheme (Alvarez-Melis and Jaakkola, 2017). Comparing our approach to other syntax aware NMT models like Eriguchi et al. (2017) and Nadejde et al. (2017) may also be of interest. A Contrastive evaluation (Sennrich, 2016) of a syntax-aware system vs. a syntax-agnostic system may also shed light on the benefits of incorporating syntax into NMT.

Acknowledgments

This work was supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI), and The Israeli Science Foundation (grant number 1555/15).

[7] "that" also functions as a determiner. We do not distinguish the two cases.

References

David Alvarez-Melis and Tommi S. Jaakkola. 2017. Tree-structured decoding with doubly recurrent neural networks. *International Conference on Learning Representations (ICLR)* .

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

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675* .

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. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 131–198.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 173–180.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 263–270.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics* 33(2):201–228.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2331–2336. https://aclweb.org/anthology/D16-1257.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 199–209. http://www.aclweb.org/anthology/N16-1024.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016a. Character-based decoding in tree-to-sequence attention-based neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*. pages 175–183.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016b. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 823–833. http://www.aclweb.org/anthology/P16-1078.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. *arXiv preprint arXiv:1702.03525* http://arxiv.org/abs/1702.03525.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 961–968.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*. Association for Computational Linguistics, Boston, Massachusetts, USA, pages 273–280.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1700–1709. http://www.aclweb.org/anthology/D13-1176.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

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

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* .

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Syntax-aware neural machine translation using CCG. *arXiv preprint arXiv:1702.01147* .

Rico Sennrich. 2016. How grammatical is character-level neural machine translation? assessing mt quality with contrastive translation pairs. *arXiv preprint arXiv:1612.04629* .

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel L"aubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 83–91. http://www.aclweb.org/anthology/W16-2209.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 371–376. http://www.aclweb.org/anthology/W16-2323.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. http://www.aclweb.org/anthology/P16-1162.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1526–1534. https://aclweb.org/anthology/D16-1159.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 299–305. http://anthology.aclweb.org/P16-2049.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215* .

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1556–1566. http://www.aclweb.org/anthology/P15-1150.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. pages 2773–2781.

P. Williams, M. Gertz, and M. Post. 2016. *Syntax-Based Statistical Machine Translation*. Morgan & Claypool publishing.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 523–530.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical mt. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 303–310.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

A Supplementary Material

Data The English side of the corpus was tokenized (into Penn treebank format) and truecased using the scripts provided in Moses (Koehn et al., 2007). We ran the BPE process on a concatenation of the source and target corpus, with 89500 BPE operations in the WMT experiment and with 45k operations in the other experiments. This resulted in an input vocabulary of 84924 tokens and an output vocabulary of 78499 tokens in the WMT16 experiment. The linearized constituency trees are obtained by simply replacing the POS tags in the parse trees with the corresponding word or subwords. The output vocabulary in the bpe2tree models includes the target subwords and the tree symbols which correspond to an opening or closing of a specific phrase type.

Hyperparameters The word embedding size was set to 500/256 and the encoder and decoder sizes were set to 1024/256 (WMT16/other experiments). For optimization we used Adadelta (Zeiler, 2012) with minibatch size of 40. For decoding we used beam search with a beam size of 12. We trained the bpe2tree WMT16 model on sequences with a maximum length of 150 tokens (the average length for a linearized tree in the training set was about 50 tokens). It was trained for two weeks on a single Nvidia TitanX GPU. The bpe2bpe WMT16 model was trained on sequences with a maximum length of 50 tokens, and with minibatch size of 80. It was trained for one week on a single Nvidia TitanX GPU. Only in the low-resource experiments we applied dropout as described in Sennrich et al. (2016a) for Romanian-English.

Human Evaluation We performed human-evaluation on the Mechnical Turk platform. Each sentence was evaluated using two annotators. For each sentence, we presented the annotators with the English reference sentence, followed by the outputs of the two systems. The German source was not shown, and the two system's outputs were shown in random order. The annotators were instructed to answer "Which of the two sentences, in your view, is a better portrayal of the the reference sentence." They were then given 6 options: "sent 1 is better", "sent 2 is better", "sent 1 is a little better", "sent 2 is a little better", "both sentences are equally good", "both sentences are equally bad". We then ignore differences between "better" and "a little better". We count as "strongly better" the cases where both annotators indicated the same sentence as better, as "weakly better" the cases were one annotator chose a sentence and the other indicated they are both good/bad. Other cases are treated as either "both good" / "both bad" or as disagreements.

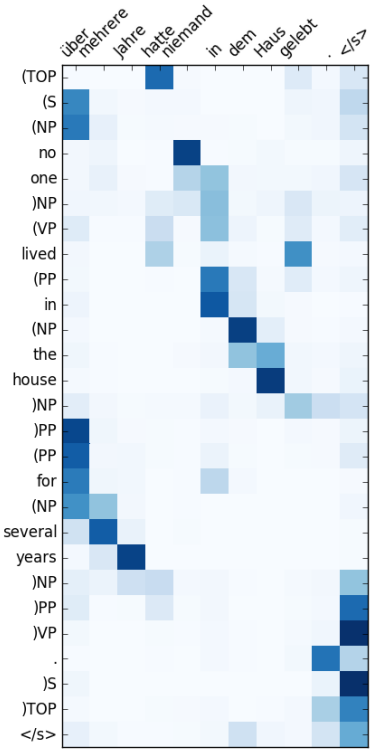
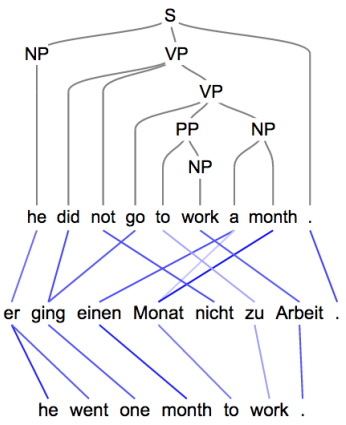
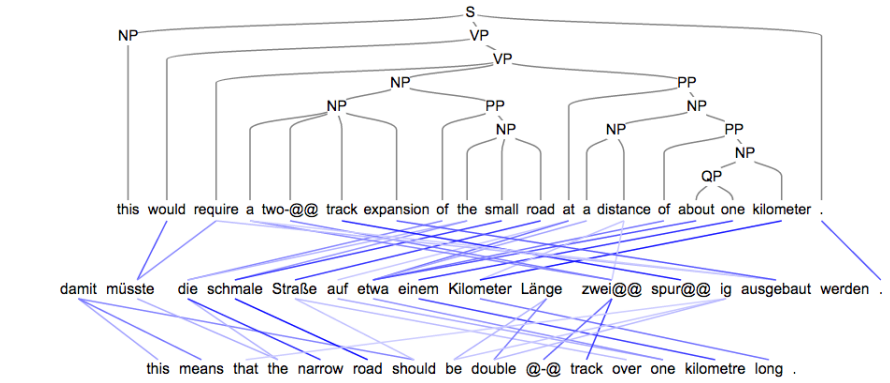
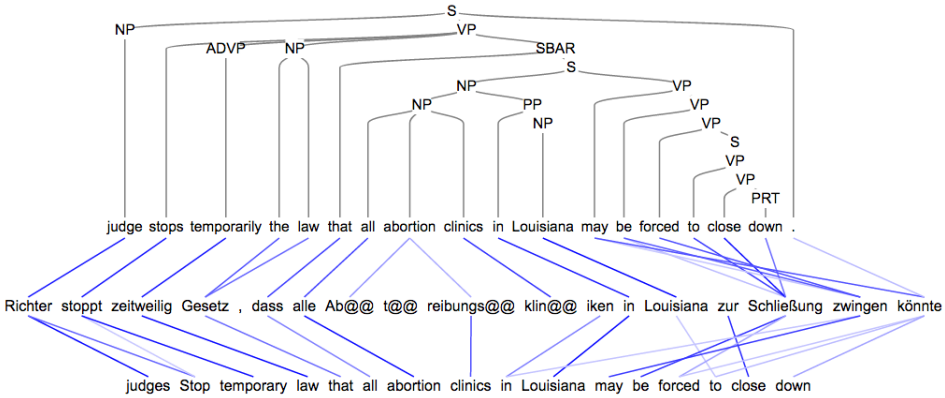


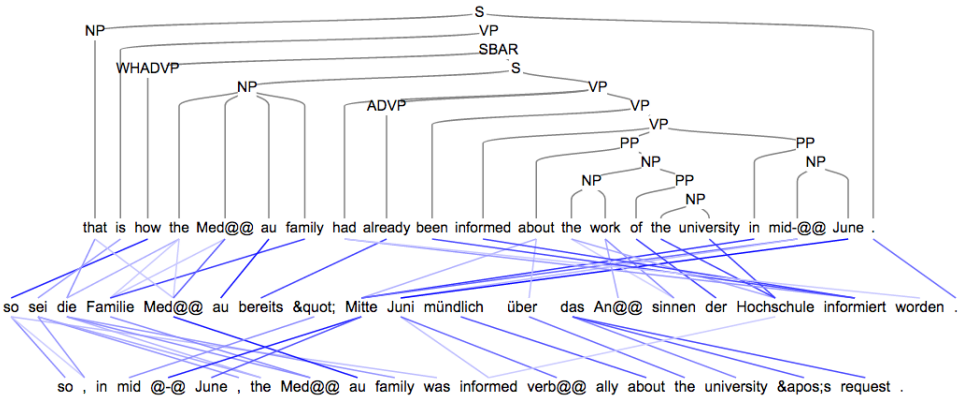
Figure 6: The attention weights for the string-to-tree translation in Figure 1

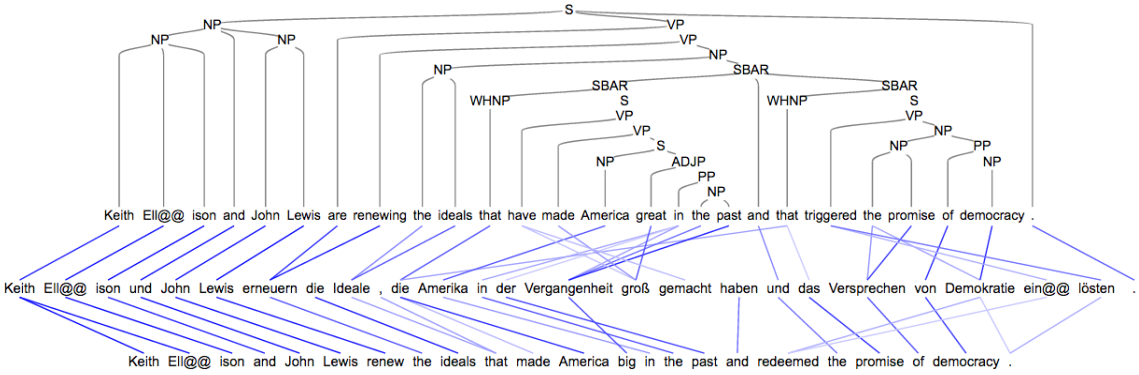
Additional Output Examples from both models, in the format of Figure 1. Notice the improved translation and alignment quality in the tree-based translations, as well as the overall high structural quality of the resulting trees. The few syntactic mistakes in these examples are attachment errors of SBAR and PP phrases, which will also challenge dedicated parsers.




this would require a two-@@ track expansion of the small road at a distance of about one kilometer .
damit müsste die schmale Straße auf etwa einem Kilometer Länge zwei@@ spur@@ ig ausgebaut werden .
this means that the narrow road should be double @-@ track over one kilometre long .


judge stops temporarily the law that all abortion clinics in Louisiana may be forced to close down .
Richter stoppt zeitweilig Gesetz , dass alle Ab@@ t@@ reibungs@@ klin@@ iken in Louisiana zur Schließung zwingen könnte
judges Stop temporary law that all abortion clinics in Louisiana may be forced to close down


that is how the Med@@ au family had already been informed about the work of the university in mid-@@ June .
so sei die Familie Med@@ au bereits " Mitte Juni mündlich über das An@@ sinnen der Hochschule informiert worden .
so , in mid @-@ June , the Med@@ au family was informed verb@@ ally about the university 's request .


Keith Ell@@ ison and John Lewis are renewing the ideals that have made America great in the past and that triggered the promise of democracy .
Keith Ell@@ ison und John Lewis erneuern die Ideale , die Amerika in der Vergangenheit groß gemacht haben und das Versprechen von Demokratie ein@@ lösten .
Keith Ell@@ ison and John Lewis renew the ideals that made America big in the past and redeemed the promise of democracy .

Chapter 5

Semantic Sentence Simplification by Splitting and Rephrasing Complex Sentences

Split and Rephrase: Better Evaluation and a Stronger Baseline

Roee Aharoni & Yoav Goldberg
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{roee.aharoni,yoav.goldberg}@gmail.com

Abstract

Splitting and rephrasing a complex sentence into several shorter sentences that convey the same meaning is a challenging problem in NLP. We show that while vanilla seq2seq models can reach high scores on the proposed benchmark (Narayan et al., 2017), they suffer from memorization of the training set which contains more than 89% of the unique simple sentences from the validation and test sets. To aid this, we present a new train-development-test data split and neural models augmented with a copy-mechanism, outperforming the best reported baseline by 8.68 BLEU and fostering further progress on the task.

1 Introduction

Processing long, complex sentences is challenging. This is true either for humans in various circumstances (Inui et al., 2003; Watanabe et al., 2009; De Belder and Moens, 2010) or in NLP tasks like parsing (Tomita, 1986; McDonald and Nivre, 2011; Jelínek, 2014) and machine translation (Chandrasekar et al., 1996; Pouget-Abadie et al., 2014; Koehn and Knowles, 2017). An automatic system capable of breaking a complex sentence into several simple sentences that convey the same meaning is very appealing.

A recent work by Narayan et al. (2017) introduced a dataset, evaluation method and baseline systems for the task, naming it "Split-and-Rephrase". The dataset includes 1,066,115 instances mapping a single complex sentence to a sequence of sentences that express the same meaning, together with RDF triples that describe their semantics. They considered two system setups: a text-to-text setup that does not use the accompanying RDF information, and a semantics-augmented setup that does. They report a BLEU score of 48.9 for their best text-to-text system, and of 78.7 for the best RDF-aware one. We focus on the text-to-text setup, which we find to be more challenging and more natural.

We begin with vanilla SEQ2SEQ models with attention (Bahdanau et al., 2015) and reach an accuracy of 77.5 BLEU, substantially outperforming the text-to-text baseline of Narayan et al. (2017) and approaching their best RDF-aware method. However, manual inspection reveal many cases of unwanted behaviors in the resulting outputs: (1) many resulting sentences are *unsupported* by the input: they contain correct facts about relevant entities, but these facts were not mentioned in the input sentence; (2) some facts are *repeated*—the same fact is mentioned in multiple output sentences; and (3) some facts are *missing*—mentioned in the input but omitted in the output.

The model learned to *memorize entity-fact pairs* instead of learning to split and rephrase. Indeed, feeding the model with examples containing entities alone without any facts about them causes it to output perfectly phrased but unsupported facts (Table 3). Digging further, we find that 99% of the simple sentences (more than 89% of the unique ones) in the validation and test sets also appear in the training set, which—coupled with the good memorization capabilities of SEQ2SEQ models and the relatively small number of distinct simple sentences—helps to explain the high BLEU score.

To aid further research on the task, we propose a more challenging split of the data. We also establish a stronger baseline by extending the SEQ2SEQ approach with a copy mechanism, which was shown to be helpful in similar tasks (Gu et al., 2016; Merity et al., 2017; See et al., 2017). On the original split, our models outperform the

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 719–724
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

	count	unique
RDF entities	32,186	925
RDF relations	16,093	172
complex sentences	1,066,115	5,544
simple sentences	5,320,716	9,552
train complex sentences	886,857	4,438
train simple sentences	4,451,959	8,840
dev complex sentences	97,950	554
dev simple sentences	475,337	3,765
test complex sentences	81,308	554
test simple sentences	393,420	4,015
% dev simple in train	99.69%	90.9%
% test simple in train	99.09%	89.8%
% dev vocab in train	97.24%	
% test vocab in train	96.35%	

Table 1: Statistics for the WEBSPLIT dataset.

best baseline of Narayan et al. (2017) by up to 8.68 BLEU, without using the RDF triples. On the new split, the vanilla SEQ2SEQ models break completely, while the copy-augmented models perform better. In parallel to our work, an updated version of the dataset was released (v1.0), which is larger and features a train/test split protocol which is similar to our proposal. We report results on this dataset as well. The code and data to reproduce our results are available on Github.[1] We encourage future work on the split-and-rephrase task to use our new data split or the v1.0 split instead of the original one.

2 Preliminary Experiments

Task Definition In the split-and-rephrase task we are given a complex sentence C, and need to produce a sequence of simple sentences $T_1, ..., T_n$, $n \geq 2$, such that the output sentences convey all and only the information in C. As additional supervision, the split-and-rephrase dataset associates each sentence with a set of RDF triples that describe the information in the sentence. Note that the number of simple sentences to generate is not given as part of the input.

Experimental Details We focus on the task of splitting a complex sentence into several simple ones *without* access to the corresponding RDF triples in either train or test time. For evaluation we follow Narayan et al. (2017) and compute the averaged individual multi-reference BLEU score for each prediction.[2] We split each prediction to sentences[3] and report the average number of simple sentences in each prediction, and the average number of tokens for each simple sentence. We train vanilla sequence-to-sequence models with attention (Bahdanau et al., 2015) as implemented in the OPENNMT-PY toolkit (Klein et al., 2017).[4] Our models only differ in the LSTM cell size (128, 256 and 512, respectively). See the supplementary material for training details and hyperparameters. We compare our models to the baselines proposed in Narayan et al. (2017). HYBRIDSIMPL and SEQ2SEQ are text-to-text models, while the other reported baselines additionally use the RDF information.

Model	BLEU	#S/C	#T/S
SOURCE	55.67	1.0	21.11
REFERENCE	–	2.52	10.93
Narayan et al. (2017)			
HYBRIDSIMPL	39.97	1.26	17.55
SEQ2SEQ	48.92	2.51	10.32
MULTISEQ2SEQ*	42.18	2.53	10.69
SPLIT-MULTISEQ2SEQ*	77.27	2.84	11.63
SPLIT-SEQ2SEQ*	78.77	2.84	9.28
This work			
SEQ2SEQ128	76.56	2.53	10.53
SEQ2SEQ256	77.48	2.57	10.56
SEQ2SEQ512	75.92	2.59	10.59

Table 2: BLEU scores, simple sentences per complex sentence (#S/C) and tokens per simple sentence (#T/S), as computed over the test set. SOURCE are the complex sentences and REFERENCE are the reference rephrasings from the test set. Models marked with * use the semantic RDF triples.

Results As shown in Table 2, our 3 models obtain higher BLEU scores then the SEQ2SEQ baseline, with up to 28.35 BLEU improvement, despite being single-layer models vs. the 3-layer models used in Narayan et al. (2017). A possible explanation for this discrepancy is the SEQ2SEQ baseline using a dropout rate of 0.8, while we use 0.3 and only apply it on the LSTM outputs. Our results are also better than the MULTISEQ2SEQ and SPLIT-MULTISEQ2SEQ models, which use explicit RDF information. We also present the macro-average[5] number of sim-

[1] https://github.com/biu-nlp/sprp-acl2018

[2] Note that this differs from "normal" multi-reference BLEU (as implemented in `multi-bleu.pl`) since the number of references differs among the instances in the test-set.

[3] Using NLTK v3.2.5 https://www.nltk.org/

[4] https://github.com/OpenNMT/OpenNMT-py commit d4ab35a

[5] Since the number of references varies greatly from one complex sentence to another, (min: 1, max: 76,283, median: 16) we avoid bias towards the complex sentences with many references by performing macro average, i.e. we first average the number of simple sentences in each reference among the

Input	Prediction
A Fortress of Grey Ice with ISBM 0-7653-0633-6 has 672 pages .	J.V. Jones authored A Fortress of Grey Ice . A Fortress of Grey Ice has 672 pages .
The address , 11 Diagonal Street is located in South Africa where the leader is Cyril Ramaphosa and some Asian South Africans live .	The address , 11 Diagonal Street is located in South Africa . The leader of South Africa is called Cyril Ramaphosa . The leader of South Africa is called Cyril Ramaphosa . The leader of South Africa is called Cyril Ramaphosa .
Alan Shepard Alan Shepard Alan Shepard	Alan Shepard is dead . Alan Shepard was a test pilot .
AFC Ajax AFC Ajax AFC Ajax	AFC Ajax 's manager is Jong Ajax . AFC Ajax N.V. own Sportpark De Toekomst .

Table 3: Predictions from a vanilla SEQ2SEQ model, illustrating unsupported facts, missing facts and repeated facts. The last two rows show inputs we composed to demonstrate that the models memorize entity-fact pairs.

ple sentences per complex sentence in the reference rephrasings (REFERENCE) showing that the SPLIT-MULTISEQ2SEQ and SPLIT-SEQ2SEQ baselines may suffer from over-splitting since the reference splits include 2.52 simple sentences on average, while the mentioned models produced 2.84 sentences.

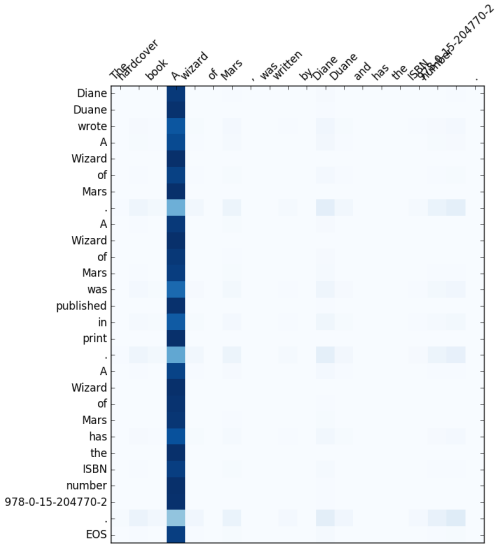


Figure 1: SEQ2SEQ512's attention weights. Horizontal: input. Vertical: predictions.

Analysis We begin analyzing the results by manually inspecting the model's predictions on the validation set. This reveals three common kinds of mistakes as demonstrated in Table 3: unsupported facts, repetitions, and missing facts. All the unsupported facts seem to be related to entities mentioned in the source sentence. Inspecting the attention weights (Figure 1) reveals a worrying trend: throughout the prediction, the model focuses heavily on the first word in of the first entity ("A wizard of Mars") while paying little attention to other cues like "hardcover", "Diane" and "the ISBN number". This explains the abundance of "hallucinated" unsupported facts: rather than learning to split and rephrase, the model learned to identify entities, and spit out a list of facts it had memorized about them. To validate this assumption, we count the number of predicted sentences which appeared as-is in the training data. We find that 1645 out of the 1693 (97.16%) predicted sentences appear verbatim in the training set. Table 1 gives more detailed statistics on the WEBSPLIT dataset.

To further illustrate the model's recognize-and-spit strategy, we compose inputs containing an entity string which is duplicated three times, as shown in the bottom two rows of Table 3. As expected, the model predicted perfectly phrased and correct facts about the given entities, although these facts are clearly not supported by the input.

3 New Data-split

The original data-split is not suitable for measuring generalization, as it is susceptible to "cheating" by fact memorization. We construct a new train-development-test split to better reflect our expected behavior from a split-and-rephrase model. We split the data into train, development and test sets by randomly dividing the 5,554 distinct complex sentences across the sets, while using the provided RDF information to ensure that:

1. Every possible RDF relation (e.g., BORNIN, LOCATEDIN) is represented in the training set (and may appear also in the other sets).

2. Every RDF triplet (a complete fact) is represented only in one of the splits.

While the set of complex sentences is still divided roughly to 80%/10%/10% as in the original split, now there are nearly no simple sentences in

references of a specific complex sentence, and then average these numbers.

	count	unique
train complex sentences	1,039,392	4,506
train simple sentences	5,239,279	7,865
dev complex sentences	13,294	535
dev simple sentences	39,703	812
test complex sentences	13,429	503
test simple sentences	41,734	879
# dev simple in train	35 (0.09%)	
# test simple in train	1 (0%)	
% dev vocab in train	62.99%	
% test vocab in train	61.67%	
dev entities in train	26/111 (23.42%)	
test entities in train	25/120 (20.83%)	
dev relations in train	34/34 (100%)	
test relations in train	37/37 (100%)	

Table 4: Statistics for the RDF-based data split

the development and test sets that appear verbatim in the train-set. Yet, every relation appearing in the development and test sets is supported by examples in the train set. We believe this split strikes a good balance between challenge and feasibility: to succeed, a model needs to learn to identify relations in the complex sentence, link them to their arguments, and produce a rephrasing of them. However, it is not required to generalize to unseen relations. [6]

The data split and scripts for creating it are available on Github.[7] Statistics describing the data split are detailed in Table 4.

4 Copy-augmented Model

To better suit the split-and-rephrase task, we augment the SEQ2SEQ models with a copy mechanism. Such mechanisms have proven to be beneficial in similar tasks like abstractive summarization (Gu et al., 2016; See et al., 2017) and language modeling (Merity et al., 2017). We hypothesize that biasing the model towards copying will improve performance, as many of the words in the simple sentences (mostly corresponding to entities) appear in the complex sentence, as evident by the relatively high BLEU scores for the SOURCE baseline in Table 2.

Copying is modeled using a "copy switch" probability $p(z)$ computed by a sigmoid over a learned composition of the decoder state, the context vector and the last output embedding. It interpolates the $p_{softmax}$ distribution over the target vocabulary and a copy distribution p_{copy} over the source sentence tokens. p_{copy} is simply the computed attention weights. Once the above distributions are computed, the final probability for an output word w is:

$$p(w) = p(z = 1)p_{copy}(w) + p(z = 0)p_{softmax}(w)$$

In case w is not present in the output vocabulary, we set $p_{softmax}(w) = 0$. We refer the reader to See et al. (2017) for a detailed discussion regarding the copy mechanism.

[6] The updated dataset (v1.0, published by Narayan et al. after this work was accepted) follows (2) above, but not (1).

[7] https://github.com/biu-nlp/sprp-acl2018

		BLEU	#S/C	#T/S
original data split	SOURCE	55.67	1.0	21.11
	REFERENCE	–	2.52	10.93
	SPLIT-SEQ2SEQ	78.77	2.84	9.28
	SEQ2SEQ128	76.56	2.53	10.53
	SEQ2SEQ256	77.48	2.57	10.56
	SEQ2SEQ512	75.92	2.59	10.59
	COPY128	78.55	2.51	10.29
	COPY256	83.73	2.49	10.66
	COPY512	87.45	2.56	10.50
new data split	SOURCE	55.66	1.0	20.37
	REFERENCE	–	2.40	10.83
	SEQ2SEQ128	5.55	2.27	11.68
	SEQ2SEQ256	5.28	2.27	10.54
	SEQ2SEQ512	6.68	2.44	10.23
	COPY128	16.71	2.0	10.53
	COPY256	23.78	2.38	10.55
	COPY512	24.97	2.87	10.04
v1.0	SOURCE	56.1	1.0	20.4
	REFERENCE	–	2.48	10.69
	COPY512	25.47	2.29	11.74

Table 5: Results over the test sets of the original, our proposed split and the v1.0 split

5 Experiments and Results

Models with larger capacities may have greater representation power, but also a stronger tendency to memorize the training data. We therefore perform experiments with copy-enhanced models of varying LSTM widths (128, 256 and 512). We train the models using the negative log likelihood of $p(w)$ as the objective. Other than the copy mechanism, we keep the settings identical to those in Section 2. We train models on the original split, our proposed data split and the v1.0 split.

Results Table 5 presents the results. On the original data-split, the COPY512 model outperforms all baselines, improving over the previous best by 8.68 BLEU points. On the new data-split, as expected, the performance degrades for all models, as they are required to generalize to sentences not seen during training. The copy-augmented models perform better than the baselines in this case as well, with a larger relative gap which can be explained by the lower lexical overlap between the train and the test sets in the new split. On the v1.0 split the results are similar to those on our split,

Input	Prediction
Madeleine L'Engle who is influenced by George Macdonald wrote " A Severed Wasp " .	A Severed Wasp was written by George Macdonald . A Severed Wasp is from the United States .
The A.C. Lumezzane has 4150 members and play in the Lega Pro League .	A.C. Lumezzane 's ground is in the Lega Pro League . A.C. Lumezzane has 4150 members .
Irish English is the official language of Ireland , which is lead by Enda Kenny and home to Adare Manor .	Adare Manor is located in Ireland . English is the language of Ireland . English is the language of Ireland . (repeated x3)

Table 6: Predictions from the COPY512 model, trained on the new data split.

in spite of it being larger (1,331,515 vs. 886,857 examples), indicating that merely adding data will not solve the task.

Analysis We inspect the models' predictions for the first 20 complex sentences of the original and new validation sets in Table 7. We mark each simple sentence as being "correct" if it contains all and only relevant information, "unsupported" if it contains facts not present in the source, and "repeated" if it repeats information from a previous sentence. We also count missing facts. Figure 2 shows the attention weights of the COPY512 model for the same sentence in Figure 1. Reassuringly, the attention is now distributed more evenly over the input symbols. On the new splits, all models perform catastrophically. Table 6 shows outputs from the COPY512 model when trained on the new split. On the original split, while SEQ2SEQ128 mainly suffers from missing information, perhaps due to insufficient memorization capacity, SEQ2SEQ512 generated the most unsupported sentences, due to overfitting or memorization. The overall number of issues is clearly reduced in the copy-augmented models.

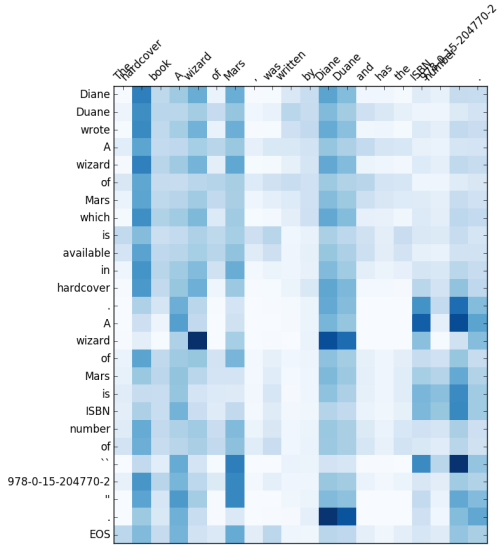


Figure 2: Attention weights from the COPY512 model for the same input as in Figure 1.

Model	unsup.	repeated	correct	missing
original split				
SEQ2SEQ128	5	4	40/49 (82%)	9
SEQ2SEQ256	2	2	42/46 (91%)	5
SEQ2SEQ512	12	2	36/49 (73%)	5
COPY128	3	4	42/49 (86%)	4
COPY256	3	2	45/50 (90%)	6
COPY512	5	0	46/51 (90%)	3
new split				
SEQ2SEQ128	37	8	0	54
SEQ2SEQ256	41	7	0	54
SEQ2SEQ512	43	5	0	54
COPY128	23	3	2/27 (7%)	52
COPY256	35	2	3/40 (7%)	49
COPY512	36	13	11/54 (20%)	43
v1.0 split				
COPY512	41	3	3/44 (7%)	51

Table 7: Results of the manual analysis, showing the number of simple sentences with unsupported facts (unsup.), repeated facts, missing facts and correct facts, for 20 complex sentences from the original and new validation sets.

6 Conclusions

We demonstrated that a SEQ2SEQ model can obtain high scores on the original split-and-rephrase task while not actually learning to split-and-rephrase. We propose a new and more challenging data-split to remedy this, and demonstrate that the cheating SEQ2SEQ models fail miserably on the new split. Augmenting the SEQ2SEQ models with a copy-mechanism improves performance on both data splits, establishing a new competitive baseline for the task. Yet, the split-and-rephrase task (on the new split) is still far from being solved. We strongly encourage future research to evaluate on our proposed split or on the recently released version 1.0 of the dataset, which is larger and also addresses the overlap issues mentioned here.

Acknowledgments

We thank Shashi Narayan and Jan Botha for their useful comments. The work was supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI), the Israeli Science Foundation (grant number 1555/15), and the German Research Foundation via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics*. Association for Computational Linguistics.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*. ACM.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany. http://www.aclweb.org/anthology/P16-1154.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics.

Tomáš Jelínek. 2014. Improvements to dependency parsing using automatic simplification of data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, Vancouver, Canada. http://aclweb.org/anthology/P17-4012.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, Vancouver. http://www.aclweb.org/anthology/W17-3204.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics* 37.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://aclweb.org/anthology/D17-1064.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merrienboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar. http://www.aclweb.org/anthology/W14-4009.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. http://www.aclweb.org/anthology/P17-1099.

Masaru Tomita. 1986. Efficient parsing for natural languagea fast algorithm for practical systems. int. series in engineering and computer science.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*. ACM.

Chapter 6

Massively Multilingual Neural Machine Translation: Towards Universal Translation

Massively Multilingual Neural Machine Translation

Roee Aharoni*
Bar Ilan University
Ramat-Gan
Israel
roee.aharoni@gmail.com

Melvin Johnson and **Orhan Firat**
Google AI
Mountain View
California
melvinp,orhanf@google.com

Abstract

Multilingual neural machine translation (NMT) enables training a single model that supports translation from multiple source languages into multiple target languages. In this paper, we push the limits of multilingual NMT in terms of the number of languages being used. We perform extensive experiments in training massively multilingual NMT models, translating up to 102 languages to and from English within a single model. We explore different setups for training such models and analyze the trade-offs between translation quality and various modeling decisions. We report results on the publicly available TED talks multilingual corpus where we show that massively multilingual many-to-many models are effective in low resource settings, outperforming the previous state-of-the-art while supporting up to 59 languages. Our experiments on a large-scale dataset with 102 languages to and from English and up to one million examples per direction also show promising results, surpassing strong bilingual baselines and encouraging future work on massively multilingual NMT.

1 Introduction

Neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014; Sutskever et al., 2014) is the current state-of-the-art approach for machine translation in both academia (Bojar et al., 2016, 2017, 2018) and industry (Wu et al., 2016; Hassan et al., 2018). Recent works (Dong et al., 2015; Firat et al., 2016a; Ha et al., 2016; Johnson et al., 2017) extended the approach to support multilingual translation, i.e. training a single model that is capable of translating between multiple language pairs.

Multilingual models are appealing for several reasons. First, they are more efficient in terms of the number of required models and model parameters, enabling simpler deployment. Another benefit is transfer learning; when low-resource language pairs are trained together with high-resource ones, the translation quality may improve (Zoph et al., 2016; Nguyen and Chiang, 2017). An extreme case of such transfer learning is zero-shot translation (Johnson et al., 2017), where multilingual models are able to translate between language pairs that were never seen during training.

While very promising, it is still unclear how far one can scale multilingual NMT in terms of the number of languages involved. Previous works on multilingual NMT typically trained models with up to 7 languages (Dong et al., 2015; Firat et al., 2016b; Ha et al., 2016; Johnson et al., 2017; Gu et al., 2018) and up to 20 trained directions (Cettolo et al., 2017) simultaneously. One recent exception is Neubig and Hu (2018) who trained many-to-one models from 58 languages into English. While utilizing significantly more languages than previous works, their experiments were restricted to many-to-one models in a low-resource setting with up to 214k examples per language-pair and were evaluated only on four translation directions.

In this work, we take a step towards practical "universal" NMT – training massively multilingual models which support up to 102 languages and with up to one million examples per language-pair simultaneously. Specifically, we focus on training "English-centric" many-to-many models, in which the training data is composed of many language pairs that contain English either on the source side or the target side. This is a realistic setting since English parallel data is widely available for many language pairs. We restrict our experiments to Transformer models (Vaswani et al., 2017) as they were shown to be very effective in recent benchmarks (Ott et al., 2018), also in

*Work carried out during an internship at Google AI.

Proceedings of NAACL-HLT 2019, pages 3874–3884
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

the context of multilingual models (Lakew et al., 2018; Sachan and Neubig, 2018).

We evaluate the performance of such massively multilingual models while varying factors like model capacity, the number of trained directions (tasks) and low-resource vs. high-resource settings. Our experiments on the publicly available TED talks dataset (Qi et al., 2018) show that massively multilingual many-to-many models with up to 58 languages to-and-from English are very effective in low resource settings, allowing to use high-capacity models while avoiding overfitting and achieving superior results to the current state-of-the-art on this dataset (Neubig and Hu, 2018; Wang et al., 2019) when translating into English.

We then turn to experiment with models trained on 103 languages in a high-resource setting. For this purpose we compile an English-centric in-house dataset, including 102 languages aligned to-and-from English with up to one million examples per language pair. We then train a single model on the resulting 204 translation directions and find that such models outperform strong bilingual baselines by more than 2 BLEU averaged across 10 diverse language pairs, both to-and-from English. Finally, we analyze the trade-offs between the number of involved languages and translation accuracy in such settings, showing that massively multilingual models generalize better to zero-shot scenarios. We hope these results will encourage future research on massively multilingual NMT.

2 Low-Resource Setting: 59 Languages

2.1 Experimental Setup

The main question we wish to answer in this work is how well a single NMT model can scale to support a very large number of language pairs. The answer is not trivial: on the one hand, training multiple language pairs together may result in transfer learning (Zoph et al., 2016; Nguyen and Chiang, 2017). This may improve performance as we increase the number of language pairs, since more information can be shared between the different translation tasks, allowing the model to learn which information to share. On the other hand, adding many language pairs may result in a bottleneck; the model has a limited capacity while it needs to handle this large number of translation tasks, and sharing all parameters between the different languages can be sub-optimal (Wang et al., 2018) especially if they are not from the same typological language family (Sachan and Neubig, 2018).

We begin tackling this question by experimenting with the TED Talks parallel corpus compiled by Qi et al. (2018)[1], which is unique in that it includes parallel data from 59 languages. For comparison, this is significantly "more multilingual" than the data available from all previous WMT news translation shared task evaluations throughout the years – the latest being Bojar et al. (2016, 2017, 2018), which included 14 languages so far.[2]

We focus on the setting where we train "English-centric" models, i.e. training on all language pairs that contain English in either the source or the target, resulting in 116 translation directions. This dataset is also highly imbalanced, with language pairs including between 3.3k to 214k sentence pairs for training. Table 9 in the supplementary material details the languages and training set sizes for this dataset. Since the dataset is already tokenized we did not apply additional preprocessing other than applying joint subword segmentation (Sennrich et al., 2016) with 32k symbols.

Regarding the languages we evaluate on, we begin with the same four languages as Neubig and Hu (2018) – Azerbeijani (Az), Belarusian (Be), Galician (Gl) and Slovak (Sk). These languages present an extreme low-resource case, with as few as 4.5k training examples for Belarusian-English. In order to better understand the effect of training set size in these settings, we evaluate on four additional languages that have more than 167k training examples each – Arabic (Ar), German (De), Hebrew (He) and Italian (It).

2.2 Model Details

Using the same data, we trained three massively multilingual models: a many-to-many model which we train using all 116 translation directions with 58 languages to-and-from English, a one-to-many model from English into 58 languages, and a many-to-one model from 58 languages into English. We follow the method of Ha et al. (2016); Johnson et al. (2017) and add a target-language

[1] github.com/neulab/word-embeddings-for-nmt

[2] Chinese, Czech, English, Estonian, Finnish, French, German, Hindi, Hungarian, Latvian, Romanian, Russian, Spanish, Turkish. According to http://www.statmt.org/wmtXX

prefix token to each source sentence to enable many-to-many translation. These different setups enable us to examine the effect of the number of translation tasks on the translation quality as measured in BLEU (Papineni et al., 2002). We also compare our massively multilingual models to bilingual baselines and to two recently published results on this dataset (Neubig and Hu (2018); Wang et al. (2019)).

Regarding the models, we focused on the Transformer in the "Base" configuration. We refer the reader to Vaswani et al. (2017) for more details on the model architecture. Specifically, we use 6 layers in both the encoder and the decoder, with model dimension set at 512, hidden dimension size of 2048 and 8 attention heads. We also applied dropout at a rate of 0.2 in the following components: on the sum of the input embeddings and the positional embeddings, on the output of each sub-layer before added to the previous layer input (residual connection), on the inner layer output after the ReLU activation in each feed-forward sub-layer, and to the attention weight in each attention sub-layer. This results in a model with approximately 93M trainable parameters. For all models we used the inverse square root learning rate schedule from Vaswani et al. (2017) with learning-rate set at 3 and 40k warmup steps. All models are implemented in Tensorflow-Lingvo (Shen et al., 2019).

In all cases we report test results for the checkpoint that performed best on the development set in terms of BLEU. For the multilingual models we create a development set that includes examples we uniformly sample from a concatenation of all the individual language pair development sets, resulting in 13k development examples per model. Another important detail regarding multilingual training is the batching scheme. In all of our multilingual models we use heterogeneous batching, where each batch contains examples which are uniformly sampled from a concatenation of all the language pairs the model is trained on. Specifically, we use batches of 64 examples for sequences shorter than 69 tokens and batches of 16 examples for longer sequences. We did not use oversampling as the dataset is relatively small.

2.3 Results

We use tokenized BLEU in order to be comparable with Neubig and Hu (2018). Table 1 shows the results of our experiments when evaluating on the same language pairs as they did. The results under "Neubig & Hu 18" are their bilingual baselines and their best many-to-one models. Their many-to-one models use similar-language-regularization, i.e. fine-tuning a pre-trained many-to-one model with data from the language pair of interest together with data from a language pair that has a typologically-similar source language and more training data (i.e. Russian and Belarusian, Turkish and Azerbaijani). The results under "Ours" are our many-to-one and many-to-many models we trained identically in terms of model architecture and hyper-parameters.

We first note that our many-to-many model out-

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Wang et al. 18	11.82	18.71	30.3	28.77	22.4
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

Table 1: X→En test BLEU on the TED Talks corpus, for the language pairs from Neubig and Hu (2018)

	Ar-En	De-En	He-En	It-En	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	27.84	30.5	**34.37**	33.64	31.59
many-to-one	25.93	28.87	30.19	32.42	29.35
many-to-many	**28.32**	**32.97**	33.18	**35.14**	**32.4**

Table 2: X→En test BLEU on the TED Talks corpus, for language pairs with more than 167k examples

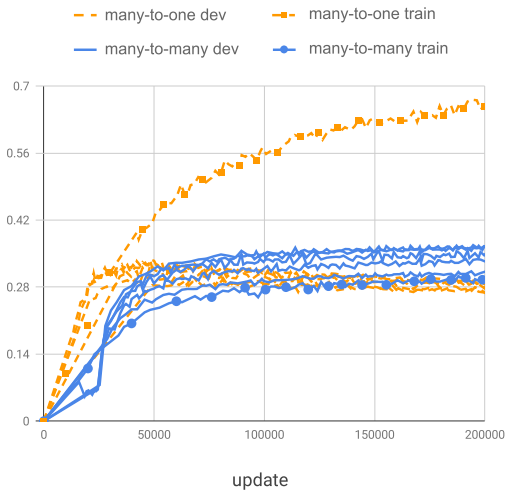


Figure 1: Development BLEU on {It,Ro,Nl,De,Ar}→En vs. training BLEU for the many-to-one and many-to-many models. Best viewed in color.

performs all other models when translating into English, with 1.82 BLEU improvement (when averaged across the four language pairs) over the best fine-tuned many-to-one models of Neubig and Hu (2018) and 2.44 BLEU improvement over our many-to-one model when averaged across the four low-resource language pairs (Table 1). This is surprising as it uses the same X→En data, model architecture and capacity as our many-to-one model, while handling a heavier burden since it also supports 58 *additional* translation tasks (*from* English *into* 58 languages). Our models also outperform the more complex models of Wang et al. (2019) which use "Soft Decoupled Encoding" for the input tokens, while our models use a simple subword segmentation.

One possible explanation is that the many-to-one model overfits the English side of the corpus as it is multi-way-parallel: in such setting the English sentences are overlapping across the different language pairs, making it much easier for the model to memorize the training set instead of generalizing (when enough capacity is available). On the other hand, the many-to-many model is trained on additional target languages other than English, which can act as regularizers for the X→En tasks, reducing such overfitting.

To further illustrate this, Figure 1 tracks the BLEU scores on the individual development sets during training for Italian (It), Romanian (Ro), Dutch (Nl), German (De) and Arabic (Ar) into English (left), together with BLEU scores on a subset of the training set for each model. We can see that while the many-to-one model degrades in performance on the development set, the many-to-many model still improves. Note the large gap in the many-to-one model between the training set BLEU and the development set BLEU, which points on the generalization issue that is not present in the many-to-many setting. We also note that our many-to-one model is on average 0.75 BLEU behind the best many-to-one models in Neubig and Hu (2018). We attribute this to the fact that their models are fine-tuned using similar-language-regularization while our model is not.

We find an additional difference between the results on the resource-scarce languages (Table 1) and the higher-resource languages (Table 2). Specifically, the bilingual baselines outperform the many-to-one models only in the higher-resource setting. This makes sense as in the low-resource setting the baselines have very few training examples to outperform the many-to-one models, while in the higher resource setting they have access to more training data. This corroborates the results of Gu et al. (2018) that showed the sensitivity of such models to similar low resource conditions and the improvements gained from using many-to-one models (however with much fewer language pairs).

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Table 3: En→X test BLEU on the TED Talks corpus

Table 3 shows the results of our massively multilingual models and bilingual baselines when evaluated out-of-English. In this case we see an opposite trend: the many-to-many model performs worse than the one-to-many model by 2.53 BLEU on average. While previous works (Wang et al., 2018; Sachan and Neubig, 2018) discuss the phenomena of quality degradation in English-to-many settings, this shows that increasing the number of *source* languages also causes additional degradation in a many-to-many model. This degradation may be due to the English-centric setting: since most of the translation directions the model is trained on are into English, this leaves less capacity for the other target languages (while still performing better than the bilingual baselines on all 8 language pairs). We also note that in this case the results are consistent among the higher and lower resource pairs – the one-to-many model is better than the many-to-many model, which outperforms the bilingual baselines in all cases. This is unlike the difference we saw in the X→ En experiments since here we do not have the multi-way-parallel overfitting issue.

2.4 Discussion

From the above experiments we learn that NMT models can scale to 59 languages in a low-resource, imbalanced, English-centric setting, with the following observations: (1) massively multilingual many-to-many models outperform many-to-one and bilingual models with similar ca-

pacity and identical training conditions when averaged over 8 language pairs into English. We attribute this improvement over the many-to-one models to the multiple target language pairs which may act as regularizers, especially in this low-resource multi-way-parallel setting that is prone to memorization. (2) many-to-many models are inferior in performance when going out-of-English in comparison to a one-to-many model. We attribute this to English being over-represented in the English-centric many-to-many setting, where it appears as a target language in 58 out of 116 trained directions, which may harm the performance on the rest of the target languages as the model capacity is limited.[3]

It is important to stress the fact that we compared the different models under *identical training conditions* and did not perform extensive hyper-parameter tuning for each setting separately. However, we believe that such tuning may improve performance even further, as the diversity in each training batch is very different between the different settings. For example, while the baseline model batches include only one language in the source and one language in the target, the many-to-many model includes 59 languages in each side with a strong bias towards English. These differences may require tailored hyper-parameter choices for each settings (i.e. different batch sizes, learning rate schedules, dropout rates etc.) which would be interesting to explore in future work.

In the following experiments we investigate whether these observations hold using (1) an even larger set of languages, and (2) a much larger, balanced training corpus that is not multi-way-parallel.

3 High-Resource Setting: 103 Languages

3.1 Experimental Setup

In this setting we scale the number of languages and examples per language pair further when training a single massively multilingual model. Since we are not aware of a publicly available resource for this purpose, we construct an in-house dataset. This dataset includes 102 language pairs which we "mirror" to-and-from English, with up to one million examples per language pair. This results in 103 languages in total, and 204 translation directions which we train simultaneously. More details about this dataset are available in Table 4, and Table 10 in the supplementary material details all the languages in the dataset.[4]

Similarly to our previous experiments, we compare the massively multilingual models to bilingual baselines trained on the same data. We tokenize the data using an in-house tokenizer and then apply joint subword segmentation to achieve an open-vocabulary. In this setting we used a vocabulary of 64k subwords rather than 32k. Since the dataset contains 24k unique characters, a 32k symbol vocabulary will consist of mostly characters, thereby increasing the average sequence length. Regarding the model, for these experiments we use a larger Transformer model with 6 layers in both the encoder and the decoder, model dimension set to 1024, hidden dimension size of 8192, and 16 attention heads. This results in a model with approximately 473.7M parameters.[5] Since the model and data are much larger in this case, we used a dropout rate of 0.1 for our multilingual models and tuned it to 0.3 for our baseline models as it improved the translation quality on the development set.

We evaluate our models on 10 languages from different typological families: *Semitic* – Arabic (Ar), Hebrew (He), *Romance* – Galician (Gl), Italian (It), Romanian (Ro), *Germanic* – German (De), Dutch (Nl), *Slavic* – Belarusian (Be), Slovak (Sk) and *Turkic* – Azerbaijani (Az) and Turkish (Tr). We evaluate both to-and-from English, where each language pair is trained on up to one million examples. As in the previous experiment, we report test results from the model that performed best in terms of BLEU on the development set.

[3]This issue may be alleviated by over-sampling the non-English-target pairs, but we leave this for future work.

[4]The average number of examples per language pair is 940k, as for 13 out of the 102 pairs we had less than one million examples available.

[5]This is larger than the Transformer "Big" configuration, which includes approximately 213M trained parameters.

# of language pairs	102
examples per pair	
min	63,879
max	1,000,000
average	940,087
std. deviation	188,194
total # of examples	95,888,938

Table 4: Training set details for the 103 langauges corpus, X→En data.

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	25.39	27.13	28.33
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

Table 5: X→En test BLEU on the 103-language corpus

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

Table 6: En→X test BLEU on the 103-language corpus

3.2 Results

Table 5 describes the results when translating into English. First, we can see that both multilingual models perform better than the baselines in terms of average BLEU. This shows that massively multilingual many-to-many models can work well in realistic settings with millions of training examples, 102 languages and 204 jointly trained directions to-and-from English. Looking more closely, we note several different behaviors in comparison to the low-resource experiments on the TED Talks corpus. First, the many-to-one model here performs better than the many-to-many model. This shows that the previous result was indeed due to the pathologies of the low-resource dataset; when the training data is large enough and not multi-way-parallel there is no overfitting in the many-to-one model, and it outperforms the many-to-many model in most cases while they are trained identically.

One particular outlier in this case is German-to-English, where the many-to-one model is 2 BLEU points below the many-to-many model. We examine the BLEU score of this language pair on its dedicated German-English development set during training in the many-to-one model and find that it highly fluctuates. We then measure the performance on the test set for this language pair by choosing the best checkpoint on the dedicated German-English development set (instead of on the mixed multilingual development set) and find it to be 38.07, which is actually *higher* in 1 BLEU than the best result of the many-to-many model. This shows that while training many languages together, there is no "silver bullet": some languages may suffer from severe interference during training (i.e. a reduction of 3 BLEU in this case, from 38.07 to 35.05) while other languages continue to improve with more updates.

Table 6 describes the results when translating out-of-English. Again, both of the massively multilingual models perform better than the baselines when averaged across the 10 evaluated language pairs, while handling up to 102 languages to-and-from English and 204 translation tasks simultaneously. In this case the results are similar to those we observed on the TED talks corpus, where the one-to-many model performs better than the many-to-many model. Again, this advantage may be due to the one-to-many model handling a smaller number of tasks while not being biased towards English in the target side like the many-to-many model.

4 Analysis

The above results show that massively multilingual NMT is indeed possible in large scale settings and can improve performance over strong bilingual baselines. However, it was shown in a somewhat extreme case with more than 100 languages trained jointly, where we saw that in some cases the joint training may harm the performance for some language pairs (i.e. German-English above). In the following analysis we would like to better understand the trade-off between the number of languages involved and the translation accuracy while keeping the model capacity and training configuration fixed.

4.1 Multilinguality & Supervised Performance

We first study the effect of varying the number of languages on the translation accuracy in a supervised setting, where we focus on many-

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

Table 7: Supervised performance while varying the number of languages involved

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17

Table 8: Zero-Shot performance while varying the number of languages involved

to-many models. We create four subsets of the in-house dataset by sub-sampling it to a different number of languages in each subset. In this way we create four additional English-centric datasets, containing 5, 25, 50 and 75 languages each to-and-from English. We make sure that each subset contains all the languages from the next smaller subsets – i.e. the 25 language subset contains the 5 language subset, the 50 language subset contains the 25 language subset and so on. We train a similar-capacity large Transformer model (with 473.7M parameters) on each of these subsets and measure the performance for each model on the 8 supervised language pairs from the smallest subset – {Arabic, French, Russian, Ukrainian}↔English. In this way we can analyze to what extent adding more languages improves or harms translation quality while keeping the model capacity fixed, testing the capacity vs. accuracy "saturation point".

Table 7 shows the results of this experiment, reporting the test results for the models that performed best on the multilingual development set. We can see that in most cases the best results are obtained using the 5-to-5 model, showing that there is indeed a trade off between the number of languages and translation accuracy when using a fixed model capacity and the same training setup. One may expect that the gaps between the different models should become smaller and even close with more updates, as the models with more languages see less examples per language in each batch, thus requiring more updates to improve in terms of BLEU. However, in our setting these gaps did not close even after the models converged, leaving 2.73 average BLEU difference between the 5-to-5 and the 103-to-103 model.

4.2 Multilinguality & Zero-Shot Performance

We then study the effect of the number of languages on zero-shot translation accuracy. Since we find zero-shot accuracy as an interesting measure for model generalization, we hypothesize that by adding more languages, the model is forced to create a more generalized representation to better utilize its capacity, which may improve zero-shot performance. We choose four language pairs for this purpose: Arabic↔French which are distant languages, and Ukrainian↔Russian which are similar. Table 8 shows the results of our models on these language pairs. For Arabic↔French the BLEU scores are very low in all cases, with the 50-to-50 and 25-to-25 models being slightly better than rest on Ar-Fr and Fr-Ar respectively. On Russian↔Ukrainian we see clear improvements when increasing the number of languages to more than five.

Figure 2 further illustrates this, showing the better generalization performance of the massively multilingual models under this zero-shot setting. While the zero-shot performance in this case is low and unstable for the 5-to-5 and 25-to-25 mod-

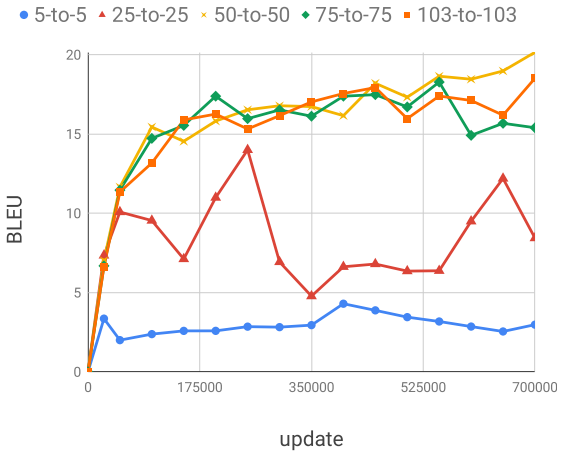


Figure 2: Zero-shot BLEU during training for Ukranian to Russian

els, it is much better for the 50-to-50, 75-to-75 and 103-to-103 models. Given these results we can say that the balance between capacity and generalization here favors the mid range 50-to-50 model, even when using models with more than 473M trained parameters. This may hint at the necessity of even larger models for such settings, which is a challenging avenue for future work. We also note that our 103 language corpus includes up to one million examples per language pair – while in real-world MT deployments, systems are trained on much more examples per pair. This again emphasizes the need for better techniques for training such massively multilingual models as we may already be hitting the capacity barrier in our setting.

5 Related Work

Dong et al. (2015) extended the NMT model of Bahdanau et al. (2014) to one-to-many translation (from English into 4 languages) by adding a dedicated decoder per target language, showing improvements over strong single-pair baselines. Firat et al. (2016a,b) proposed many-to-many models (with up to 6 languages) by using separate encoders and decoders per language while sharing the attention mechanism. They also introduced the notion of zero-resource translation, where they use synthetic training data generated through pivoting to train translation directions without available training data. Ha et al. (2016) and Johnson et al. (2017) proposed to use a shared encoder-decoder-attention model for many-to-many translation (with up to 7 languages in the latter). In order to determine the target language in such scenarios they proposed adding dedicated target-language symbols to the source. This method enabled zero-shot translation, showing the ability of the model to generalize to unseen pairs.

Recent works propose different methods for parameter sharing between language pairs in multilingual NMT. Blackwood et al. (2018) propose sharing all parameters but the attention mechanism and show improvements over sharing all parameters. Sachan and Neubig (2018) explore sharing various components in self-attentional (Transformer) models. Lu et al. (2018) add a shared "interlingua" layer while using separate encoders and decoders. Zaremoodi et al. (2018) utilize recurrent units with multiple blocks together with a trainable routing network. Platanios et al. (2018) propose to share the entire network, while using a contextual parameter generator that learns to generate the parameters of the system given the desired source and target languages. Gu et al. (2018) propose a "Universal Language Representation" layer together with a Mixture-of-Language-Experts component to improve a many-to-one model from 5 languages into English.

While the mentioned studies provide valuable contributions to improving multilingual models, they apply their models on only up to 7 languages (Johnson et al., 2017) and 20 trained directions (Cettolo et al., 2017) in a single model, whereas we focus on scaling NMT to much larger numbers of languages and trained directions. Regarding massively multilingual models, Neubig and Hu (2018) explored methods for rapid adaptation of NMT to new languages by training multilingual models on the 59-language TED Talks corpus and fine-tuning them using data from the new languages. While modeling significantly more languages than previous studies, they only train many-to-one models, which we show are inferior in comparison to our proposed massively multilingual many-to-many models when evaluated into English on this dataset.

Tiedemann (2018) trained an English-centric many-to-many model on translations of the bible including 927 languages. While this work pointed to an interesting phenomena in the latent space learned by the model where it clusters representations of typologically-similar languages together, it did not include any evaluation of the produced translations. Similarly, Malaviya et al. (2017) trained a many-to-English system including 1017 languages from bible translations, and used it to infer typological features for the different languages (without evaluating the translation quality). In another relevant work, Artetxe and Schwenk (2018) trained an NMT model on 93 languages and used the learned representations to perform cross-lingual transfer learning. Again, they did not report the performance of the translation model learned in that massively multilingual setting.

6 Conclusions and Future Work

We showed that NMT models can successfully scale to 102 languages to-and-from English with 204 trained directions and up to one million examples per direction. Such models improve the translation quality over similar single-pair base-

lines when evaluated to and from English by more than 2 BLEU when averaged over 10 diverse language pairs in each case. We show a similar result on the low-resource TED Talks corpus with 59 languages and 116 trained directions. We analyze the trade-offs between translation quality and the number of languages involved, pointing on capacity bottlenecks even with very large models and showing that massively multilingual models can generalize better to zero-shot settings.

We hope this work will encourage future research on massively multilingual NMT, enabling easier support for systems that can serve more people around the globe. There are many possible avenues for future work, including semi-supervised learning in such settings, exploring ways to reduce the performance degradation when increasing the number of languages, or using such models for multilingual transfer learning (McCann et al., 2017; Eriguchi et al., 2018; Artetxe and Schwenk, 2018). Understanding and improving zero-shot performance in such scenarios is also a promising direction for future work.

Acknowledgments

We would like to thank the Google Brain and Google Translate teams for their useful inputs and discussions. We would also like to thank the entire Lingvo development team for their foundational contributions to this project.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*.

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

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, et al. 2016. Findings of the 2016 conference on machine translation. In *ACL 2016 FIRST CONFERENCE ON MACHINE TRANSLATION (WMT16)*, pages 131–198.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*.

Mauro Cettolo, Federico Marcello, Bentivogli Luisa, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. 2018. Zero-shot cross-lingual classification using multilingual neural machine translation. *arXiv preprint arXiv:1809.04686*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. 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*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Hany Hassan, Anthony Aue, Chang Chen, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Melvin Johnson, Mike Schuster, Quoc V Le, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Belgium, Brussels.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proc. IJCNLP*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Ye Qi, Sachan Devendra, Felix Matthieu, Padmanabhan Sarguna, and Neubig Graham. 2018. When and why are pre-trained word embeddings useful for neural machine translation. In *HLT-NAACL*.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Belgium, Brussels.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, volume 1, pages 1715–1725.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2018. Emerging language spaces learned from massively multilingual corpora. *arXiv preprint arXiv:1802.00273*.

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

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Yonghui Wu, Mike Schuster, Zhifeng Chen, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

A Supplementary Material

Language	Train set size
Arabic	214111
Hebrew	211819
Russian	208458
Korean	205640
Italian	204503
Japanese	204090
Chinese-Taiwan	202646
Chinese-China	199855
Spanish	196026
French	192304
Portuguese-Brazil	184755
Dutch	183767
Turkish	182470
Romanian	180484
Polish	176169
Bulgarian	174444
Vietnamese	171995
German	167888
Persian	150965
Hungarian	147219
Serbian	136898
Greek	134327
Croatian	122091
Ukrainian	108495
Czech	103093
Thai	98064
Indonesian	87406
Slovak	61470
Swedish	56647
Portuguese	51785
Danish	44940
Albanian	44525
Lithuanian	41919
Macedonian	25335
Finnish	24222
Burmese	21497
Armenian	21360
French-Canadian	19870
Slovenian	19831
Hindi	18798
Norwegian	15825
Kannada	13193
Estonian	10738
Kurdish	10371
Galician	10017
Marathi	9840
Mongolian	7607
Esperanto	6535
Tamil	6224
Urdu	5977
Azerbaijani	5946
Bosnian	5664
Chinese	5534
Malay	5220
Basque	5182
Bengali	4649
Belarusian	4509
Kazakh	3317

Table 9: Language pairs in the TED talks dataset (58 languages, paired with English) with the train-set size for each pair.

Languages	
Afrikaans	Laothian
Albanian	Latin
Amharic	Latvian
Arabic	Lithuanian
Armenian	Luxembourgish*
Azerbaijani	Macedonian
Basque	Malagasy
Belarusian	Malay
Bengali	Malayalam
Bosnian	Maltese
Bulgarian	Maori
Burmese	Marathi
Catalan	Mongolian
Cebuano	Nepali
Chichewa*	Norwegian
Chinese	Pashto
Corsican*	Persian
Croatian	Polish
Czech	Portuguese
Danish	Punjabi
Dutch	Romanian
Esperanto	Russian
Estonian	Samoan*
Finnish	Scots Gaelic*
French	Serbian
Frisian	Sesotho
Galician	Shona*
Georgian	Sindhi*
German	Sinhalese
Greek	Slovak
Gujarati	Slovenian
Haitian Creole	Somali
Hausa*	Spanish
Hawaiian*	Sundanese
Hebrew	Swahili
Hindi	Swedish
Hmong*	Tagalog
Hungarian	Tajik*
Icelandic	Tamil
Igbo	Telugu
Indonesian	Thai
Irish	Turkish
Italian	Ukrainian
Japanese	Urdu
Javanese	Uzbek
Kannada	Vietnamese
Kazakh	Welsh
Khmer	Xhosa
Korean	Yiddish
Kurdish	Yoruba*
Kyrgyz	Zulu

Table 10: Language pairs in the in-house dataset (102 languages, paired with English). For languages marked with * we had less than 1M examples, while for the rest we used exactly 1M.

Chapter 7

Unsupervised Domain Clusters in Pretrained Language Models

Unsupervised Domain Clusters in Pretrained Language Models

Roee Aharoni[1] **& Yoav Goldberg**[1,2]
[1] Computer Science Department, Bar Ilan University
[2] Allen Institute for Artificial Intelligence
first.last@gmail.com

Abstract

The notion of "in-domain data" in NLP is often over-simplistic and vague, as textual data varies in many nuanced linguistic aspects such as topic, style or level of formality. In addition, domain labels are many times unavailable, making it challenging to build domain-specific systems. We show that massive pre-trained language models implicitly learn sentence representations that cluster by domains without supervision – suggesting a simple data-driven definition of domains in textual data. We harness this property and propose domain data selection methods based on such models, which require only a small set of in-domain monolingual data. We evaluate our data selection methods for neural machine translation across five diverse domains, where they outperform an established approach as measured by both BLEU and by precision and recall of sentence selection with respect to an oracle.

1 Introduction

It is common knowledge in modern NLP that using large amounts of high-quality training data is a key aspect in building successful machine-learning based systems. For this reason, a major challenge when building such systems is obtaining data in the domain of interest. But what defines a domain? Natural language varies greatly across topics, styles, levels of formality, genres and many other linguistic nuances (van der Wees et al., 2015; van der Wees, 2017; Niu et al., 2017). This overwhelming diversity of language makes it hard to find the right data for the task, as it is nearly impossible to well-define the exact requirements from such data with respect to all the aforementioned aspects. On top of that, domain labels are usually unavailable – e.g. in large-scale web-crawled data like Common Crawl[1] which was recently used to train state-of-the-art pretrained language models for various tasks (Raffel et al., 2019).

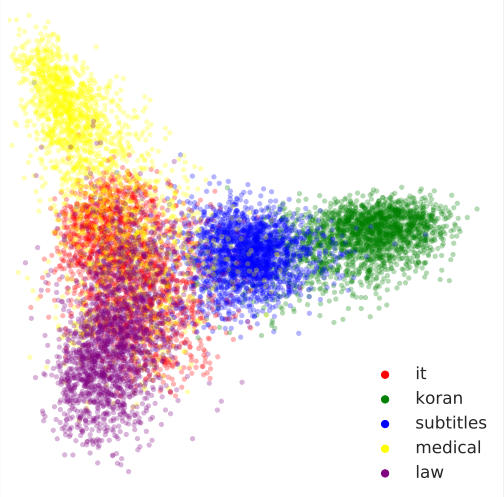


Figure 1: A 2D visualization of average-pooled BERT hidden-state sentence representations using PCA. The colors represent the domain for each sentence.

Domain data selection is the task of selecting the most appropriate data for a domain from a large corpus given a smaller set of in-domain data (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013; Silva et al., 2018). In this work, we propose to use the recent, highly successful self-supervised pre-trained language models, e.g. Devlin et al. (2019); Liu et al. (2019) for domain data selection. As pretrained LMs demonstrate state-of-the-art performance across many NLP tasks after being trained on massive amounts of data, we hypothesize that the robust representations they learn can be useful for mapping sentences to domains in an unsupervised, data-driven approach. We show that these models indeed learn to cluster sentence representations to domains without further supervision (e.g. Figure 1), and quantify this phenomenon by fitting Gaussian Mixture Models (GMMs) to the learned representations and measuring the purity of the resulting clustering. We then propose meth-

[1] https://commoncrawl.org/

ods to leverage these emergent domain clusters for domain data selection in two ways:

- Via distance-based retrieval in the sentence embedding space induced by the pretrained language model.
- By fine-tuning the pretrained language model for binary classification, where positive examples are from the domain of interest.

Our methods enable to select relevant data for the task while requiring only a small set of monolingual in-domain data. As they are based solely on the representations learned by self-supervised LMs, they do not require additional domain labels which are usually vague and over-simplify the notion of domain in textual data. We evaluate our method on data selection for neural machine translation (NMT) using the multi-domain German-English parallel corpus composed by Koehn and Knowles (2017). Our data selection methods enable to train NMT models that outperform those trained using the well-established cross-entropy difference method of Moore and Lewis (2010) across five diverse domains, achieving a recall of more than 95% in all cases with respect to an oracle that selects the "true" in-domain data.

Our contributions in this work are as follows. First, we show that pre-trained language models are highly capable of clustering textual data to domains with high accuracy in a purely unsupervised manner. Second, we propose methods to select in-domain data based on this property using vector-space retrieval and positive-unlabeled fine-tuning of pretrained language models for binary classification. Third, we show the applicability of our proposed data selection methods on a popular benchmark for domain adaptation in machine translation. An additional contribution is a new, improved data split we create for this benchmark, as we point on issues with previous splits used in the literature. We hope this work will encourage more research on understanding the data landscape in NLP, enabling to "find the right data for the task" in the age of massive models and diverse data sources.

2 Emerging Domain Clusters in Pretrained Language Models

Motivation The proliferation of massive pretrained neural language models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) has enabled great progress on many NLP benchmarks (Wang et al., 2018, 2019a). Larger and larger models trained on billions of tokens of raw text are released in an ever-increasing pace (Raffel et al., 2019), enabling the NLP community to fine-tune them for the task of interest. While many works tried to "probe" those models for the morphological, syntactic and semantic information they capture (Tenney et al., 2019; Goldberg, 2019; Clark et al., 2019), an important aspect of language remained overlooked in this context – the *domain* the data comes from, often referred to as the "data distribution".

The definition of domain is many times vague and over-simplistic (e.g. "medical text" may be used for biomedical research papers and for clinical conversations between doctors and patients, although the two vary greatly in topic, formality etc.). A common definition treats a domain as a data source: "a domain is defined by a corpus from a specific source, and may differ from other domains in topic, genre, style, level of formality, etc." (Koehn and Knowles, 2017). We claim that a more data-driven definition should take place, as different data sources may have sentences with similar traits and vice versa - a single massive web-crawled corpus contains texts in numerous styles, topics and registers. Our analysis in Section 2 shows examples for such cases, e.g. a sentence discussing "Viruses and virus-like organisms" in a legal corpus.

Unsupervised Domain Clustering We hypothesize that massive pretrained LMs can learn representations that cluster to domains, as texts from similar domains will appear in similar contexts. We test this hypothesis across several large, publicly-available pretrained LMs; we explore both masked-language-models (MLMs) and auto-regressive LMs.

Method We encode multi-domain data at the sentence level into vector representations. We then cluster these vector representations for each model using a Gaussian Mixture Model (GMM) with k pre-defined clusters. In all cases, to create a sentence representation we perform average pooling of the last hidden state (before the softmax layer) for each token in the sentence.[2] To accelerate the clustering process and enable visualization we also experiment with performing dimensionality reduction with PCA over the sentence vectors before

[2]Using the penultimate layer or others may also be a valid choice; we leave this for future work.

	k=5	k=10	k=15
Random	15.08 (±0.0)	16.77 (±0.0)	17.78 (±0.0)
LDA	24.31 (±0.99)	26.73 (±2.19)	30.79 (±2.97)

	with PCA (n=50)			without PCA		
	k=5	k=10	k=15	k=5	k=10	k=15
word2vec	53.65 (±0.79)	68.14 (±2.58)	73.44 (±0.68)	45.93	65.80	76.26
BERT-base	**87.66 (±0.24)**	88.02 (±1.10)	88.37 (±0.66)	**85.74**	85.08	86.37
BERT-large	85.64 (±6.13)	87.61 (±0.26)	89.07 (±0.53)	68.56	**86.53**	86.99
DistillBERT	83.68 (±7.14)	86.31 (±0.86)	87.53 (±0.85)	79.00	86.42	**88.14**
RoBERTa-base	79.05 (±0.10)	86.39 (±0.90)	86.51 (±0.28)	70.21	80.35	81.49
RoBERTa-large	80.61 (±0.33)	**89.04 (±0.15)**	**89.94 (±0.23)**	69.88	81.07	85.91
GPT-2	70.30 (±0.05)	84.76 (±0.30)	82.56 (±1.29)	37.82	39.02	41.45
XLNet	55.72 (±0.69)	68.17 (±3.93)	72.65 (±1.92)	30.36	32.96	48.55

Table 1: Unsupervised domain clustering as measured by purity for the different models. Best results are marked in bold for each setting.

clustering them. We experiment with k in 5, 10 and 15 to test how adding flexibility would improve the domain clustering accuracy.

Models and Baselines For MLM-based models we use BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019) and RoBERTa (Liu et al., 2019) (in both the base and large versions). For autoregressive models we use GPT-2 (Radford et al., 2018) and XLNet (Yang et al., 2019). In all cases we use the implementations from the HuggingFace Transformers toolkit (Wolf et al., 2019). We also evaluated three additional, simpler baselines. The first is using representations from word2vec (Mikolov et al., 2013), where we average-pooled the word vectors for the tokens that were present in the model vocabulary. The second is using Latent Dirichlet Allocation (LDA, Blei et al., 2003), which is a classic approach to unsupervised clustering of text.[3] We also report results for a baseline which assigns sentences by sampling randomly from a uniform distribution over the clusters.

Evaluation To evaluate the unsupervised domain clustering we used the multi-domain corpus proposed by Koehn and Knowles (2017) which includes textual data in five diverse domains: subtitles[4], medical text (PDF documents from the European Medicines Agency), legal text (legislative text of the European Union), translations of the Koran, and IT-related text (manuals and localization files of open-source software). See more details on this dataset in Section 3.1. We used 2000 distinct sentences from each domain. To evaluate whether the resulting clusters indeed capture the domains the data was drawn from we measure the purity metric, where we assign each unsupervised cluster with the most common domain in the sentences assigned to that cluster, and then compute the accuracy according to this majority-based assignment. In cases where randomness is involved we run each experiment five times with different initializations and report the mean and variance of the purity metric for each model.

Results and Discussion As can be seen in Table 1, pre-trained language models are indeed highly capable of generating sentence representations that cluster by domains, resulting in up to 87.66%, 89.04% and 89.94% accuracy when using k=5, k=10 and k=15 clusters, respectively, across 10,000 sentences in 5 domains. We find these scores remarkably high given that our average-pooling strategy is very straight-forward and that no domain-supervision was involved in the process of learning the pre-trained representations. Figure 2 also demonstrates the quality of the obtained clusters in 2D using the BERT-base model, where the ellipses describe the mean and variance parameters learned

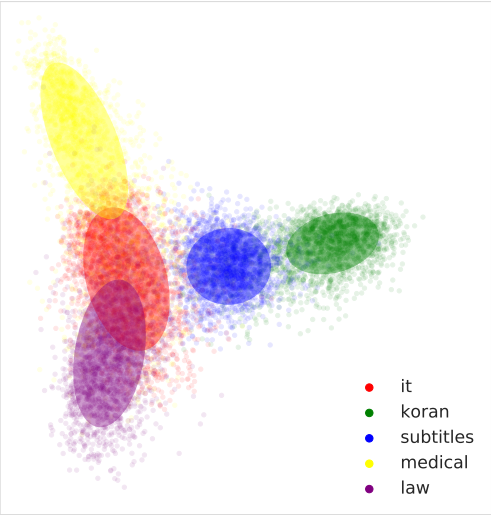


Figure 2: A 2D visualization of the unsupervised GMM clustering for the same sentences as in Figure 1.

[3] We used the LDA implementation provided in the Gensim toolkit: https://radimrehurek.com/gensim/

[4] From http://www.opensubtitles.org/

for each cluster by the GMM with $k = 5$.[5]

We note that some classes of models did better than others: while all vector-based models did far better than the random and LDA baselines, the MLM-based models dominated in all cases over word2vec and the auto-regressive models. This may be explained by the fact that the MLM-based models use the entire sentence context when generating the representations for each token, while the auto-regressive models only use the past context, and word2vec uses a limited window context. Using PCA improved performance in most cases and especially for the auto-regressive models, although the results for the MLMs remain high in both cases – suggesting that these models encode the information very differently.

Analysis As can be seen in Figure 2, in some areas the domains are somewhat overlapping in the embedding space, which may lead to outlier cases where examples from one domain are assigned to a cluster of a another domain. We plot a confusion matrix (Figure 3) to analyze this further based on the clustering with BERT-base and k=5. We first note that the outlier sentences are much shorter than the average sentence length in the corpus (11.62 tokens on average for outliers vs. 20.5 tokens on average in general). This makes sense as shorter sentences contain less information, making it harder to assign them to an appropriate cluster. Table 2 shows examples of outlier sentences, assigned to clusters of domains different from their originating domain. We can see that in many cases the assignments are sensible – for example for sentences originating from the subtitles corpus, a sentence that mentions "great priest" is assigned to the Koran cluster, a sentence that mentions "The International Criminal Court in The Hague" is assigned to the Law cluster, a sentence that mentions "the virus" is assigned to the Medical cluster and so on. This strengthens our claim that defining domains based on the corpus they originated from may be over-simplistic, and using a more data-driven approach may enable to find better domain assignments across different corpora.

The domain that attracted the largest number

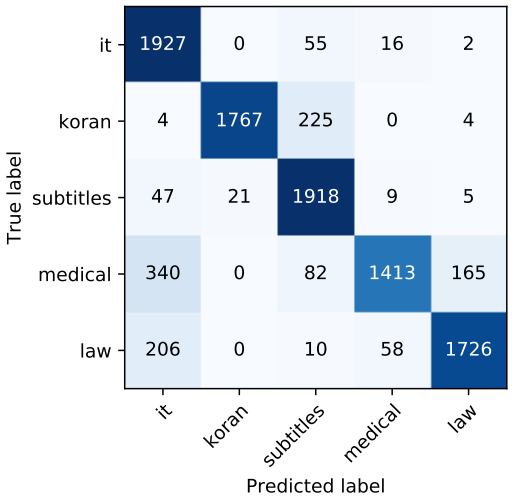


Figure 3: A confusion matrix for clustering with k=5 using BERT-base.

[5]Similar visualizations for additional models are available in the supplementary material.

Subtitles assigned to Koran	**Subtitles assigned to Medical**
I am Spa'am, high priest of the boars.	Oxygen supply at 50%.
Joseph, go in peace, and the Lord be with you.	Or it can help her walk again if the virus is kept in check with this.
Subtitles assigned to IT	**Subtitles assigned to Law**
Push it up to the front of the screen.	Statutes, transcripts, redacted immunity agreements.
Polyalloy requires programming to take permanent form.	The Security Council therefore must press for his immediate referral to the International Criminal Court in The Hague.
Law assigned to Medical	**Law assigned to IT**
- Viruses and virus-like organisms	"INFORMATION SOCIETY STATISTICS
where the glucose content is equal to or less than the fructose content.	This document must be attached to the certificate and field with it, except where there is a computerised checking system.
Medical assigned to Law	**Medical assigned to IT**
This will be introduced by a Regulation adopted by the European Commission.	An updated and improved version of the CD-ROM was issued to all subscribers during the first half of the year.
The marketing authorisation was renewed on 22 May 2002 and 22 May 2007.	- All tables will be based on generic and not product-specific data.
IT assigned to Medical	**IT assigned to Subtitles**
R65: Harmful: may cause lung damage if swallowed	At the end we say good bye.
Automatic Red-Eye Removal	What would you like to do for your next shot?

Table 2: Sentences from one domain which were assigned to a cluster of another domain by the BERT-based clustering, k=5.

of outliers is the IT domain cluster, with 597 sentences assigned to it from other domains. Looking more closely we find that more than half of these sentences (340 out of 597) included numbers (e.g. "34% 25% 34%" (from medical), "(b) reference number 20 is deleted;" (from law), "(Command of Prostration # 1)" (from Koran) or "The message, R2." (from subtitles)). As numbers appear in many different contexts, they may be harder to assign to a specific domain by the context-aware language models in such short sentences. The second largest attractor of outliers is the Subtitles cluster, with 372 sentences assigned to it from other domains. We find that most of these sentences contain personal pronouns or question marks (228 out of 372, 61.2%) while the ratio of such sentences in the entire corpus is only 40%. Examples include "Why did *you* choose the name & amarok;?" (from IT), or "What is Avonex?" (from Medical). This may be expected as the subtitles corpus mainly includes transcriptions of spoken, conversational language, and "conversation tends to have more verbs, more personal pronouns, and more questions" (Conrad and Biber, 2005). Another possible reason for the subtitles domain to attract outliers is the fact that this is the least-topical cluster: movies and TV series may discuss diverse topics, unlike medical, religious, legal and technical texts that may have a more cohesive topic.

3 Neural Machine Translation in a Multi-Domain Scenario

As we showed that pre-trained language models are indeed very useful in clustering sentence representations by domains in an unsupervised manner, we now seek to harness this property for a downstream task – domain data selection for machine translation. Domain data selection is the task of selecting examples from a large corpus which are as close as possible to the domain of interest, given a smaller set of in-domain examples. The selected examples can be used to either (1) train a domain-specific model from scratch (Axelrod et al., 2011), (2) fine-tune a pre-trained general-domain model (Silva et al., 2018), or (3) prioritize data for annotation as in an Active-Learning framework, if only monolingual data is available (Haffari et al., 2009). To demonstrate the need for domain data selection and set the stage for our data selection experiments, we perform preliminary experiments with NMT in a multi-domain scenario.

	Original	New Split
Medical	1,104,752	248,099
Law	715,372	467,309
IT	378,477	222,927
Koran	533,128	17,982
Subtitles	22,508,639	14,458,058

Table 3: Number of training examples for each domain in the original split (Müller et al., 2019) and in our split.

3.1 Multi-Domain Dataset

To simulate a diverse multi-domain setting we use the dataset proposed in Koehn and Knowles (2017), as it was recently adopted for domain adaptation research in NMT (Hu et al., 2019; Müller et al., 2019; Dou et al., 2019a,b). The dataset includes parallel text in German and English from five diverse domains (Medical, Law, Koran, IT, Subtitles; as discussed in Section 2), available via OPUS (Tiedemann, 2012; Aulamo and Tiedemann, 2019).

In a preliminary analysis of the data we found that in both the original train/dev/test split by Koehn and Knowles (2017) and in the more recent split by Müller et al. (2019) there was overlap between the training data and the dev/test data.[6] Fixing these issues is important, as it may affect the conclusions one draws from experiments with this dataset. For example, as overlapping development sets favor memorization of the training set, one may choose checkpoints and report results on over-fitting models. This is especially relevant with neural sequence-to-sequence models, as they are highly susceptible to memorization (Aharoni and Goldberg, 2018) and hallucination (Lee et al., 2018), as confirmed by Müller et al. (2019).

To create a better experimental setting to test generalization within and across domains, we create a new data split where we ensure that no such overlap between the training, development and test sets occur. We started from the split of Müller et al. (2019) as it included newer versions of some of the datasets.[7] Furthermore, we did not allow more than one translation of a given source or target sentence, as such cases were very frequent in the dataset and usually stand for duplicate sentence pairs (See Table 3). For example, applying this filtering reduced the size of the Koran corpus from 533,128 sentences to only 17,982 sentences. Finally, following Müller et al. (2019) we cap the subtitles corpus to 500,000 sentence pairs as it is much larger than the rest. We make the new split publicly available and

[6] More details are available in the supplementary material.

[7] Their dataset is available in: https://github.com/ZurichNLP/domain-robustness

hope it will enable better future experimentation on this important subject.[8]

3.2 Cross-Domain Experiments

Experimental Setup We follow Hu et al. (2019) and train domain-specific models for all domains. We then evaluate each model across the different domain test sets, enabling us to understand the effect of different domains on the downstream MT performance and to set up strong baselines for data selection experiments. We also train a general-domain model using the available data from all domains, as it is also a common approach in multi-domain scenarios (Müller et al., 2019). In all experiments we use a similar Transformer (Vaswani et al., 2017) model, and only control for the training data. More details on the exact training and hyperparameter settings for the NMT models are available in the supplementary material.

Results The results for the cross-domain evaluation are available in Table 4. In most cases, the best results for each domain are obtained by training on the in-domain data. Training on all the available data helped mostly for the Koran test set. This is expected as the training data for this domain is considerably smaller than the training data for rest of the domains (Table 3). We can also see that more data is not necessarily better (Gascó et al., 2012): while the subtitles corpus is the largest of all 5 and includes 500,000 sentence pairs, it is second to last in performance as measured by the average BLEU across all test sets.

Cross-Domain BLEU vs. Cluster Proximity An interesting observation can be made with respect to the visual analysis of the domain clusters as depicted in Figure 2: as the Medical cluster (in Yellow), Law cluster (in Purple) and IT cluster (in Red) are close to each other in the embedding space, their cross-domain BLEU scores are also higher. For example, note how in the results for the Medical domain-specific model (first row in Table 4), the BLEU scores on the Law and IT test sets are much higher in comparison to those on the Koran and Subtitles test sets, which clusters are farther away in the visualized embedding space. Similarly, as the Subtitles cluster (Blue) is closer to the Koran cluster (Green), the highest cross-domain BLEU score on the Koran test set is from the Subtitles model. This suggests that such preliminary visual analysis can be a useful tool for understanding the relationship between diverse datasets, and motivates the use of pre-trained language model representations for domain data selection in MT.

[8] https://github.com/roeeaharoni/unsupervised-domain-clusters

	Medical	Law	Koran	IT	Subtitles
Medical	**56.5**	18.3	1.9	11.4	4.3
Law	21.7	**59**	2.7	13.1	5.4
Koran	0.1	0.2	15.9	0.2	0.5
IT	14.9	9.6	2.8	**43**	8.6
Subtitles	7.9	5.5	6.4	8.5	27.3
All	53.3	57.2	**20.9**	42.1	**27.6**

Table 4: SacreBLEU (Post, 2018) scores of our baseline systems on the test sets of the new data split. Each row represents the results from one model on each test set. The best result in each column is marked in bold.

4 Domain Data Selection with Pretrained Language Models

As shown in the previous section, using the right data is critical for achieving good performance on an in-domain test set, and more data is not necessarily better. However, in real-world scenarios, the availability of data labeled by domain is limited, e.g. when working with large scale, web-crawled data. In this section we focus on a data-selection scenario where only a very small number of in-domain sentences are used to select data from a larger unlabeled parallel corpus. An established method for data selection was proposed by Moore and Lewis (2010), which was also used in training the winning systems in WMT 2019 (Ng et al., 2019; Barrault et al., 2019). This method compares the cross-entropy, according to domain-specific and non-domain-specific language models, for each candidate sentence for selection. The sentences are then ranked by the cross-entropy difference, and only the top sentences are selected for training.

While the method by Moore and Lewis (2010) is tried-and-true, it is based on simple n-gram language models which cannot generalize beyond the n-grams that are seen in the in-domain set. In addition, it is restricted to the in-domain and general-domain datasets it is trained on, which are usually small. On the contrary, pre-trained language models are trained on massive amounts of text, and, as we showed through unsupervised clustering, learn representations with domain-relevant information. In the following sections, we investigate whether this property of pretrained language models makes them useful for domain data selection.

4.1 Methods

We propose two methods for domain data selection with pretrained language models.

Domain-Cosine In this method we first compute a query vector, which is the element-wise average over the vector representations of the sentences in the small in-domain set. We use the same average-pooling approach as described in Section 2. We then retrieve the most relevant sentences in the training set by computing the cosine similarity of each sentence with this query vector and ranking the sentences accordingly.

Domain-Finetune It is now common knowledge that pretrained language models are especially useful when fine-tuned for the task of interest in an end-to-end manner. In this method we fine-tune the pretrained LM for binary classification, where we use the in-domain sentences as positive examples, and randomly sampled general-domain sentences as negative examples. We then apply this classifier on the general-domain data and pick the sentences that are classified as positive as in-domain, or choose the top-k sentences as ranked by the classifier output distribution.

Negative Sampling with Pre-ranking One problem that may rise in this case is that unlabeled in-domain sentences from the general-domain data may be sampled as negative examples and deteriorate the classifier performance. To alleviate this issue, we perform a biased sampling of negative examples. We first rank the general-domain data using the Domain-Cosine method, and then sample negative examples under a certain threshold in the ranking (in our experiments we sampled from the bottom two-thirds). Table 5 shows an ablation for pre-ranking, measuring precision, recall and F1 for binary classification on a held-out set for each domain. When not using pre-ranking, as the training data for the domain is larger, the precision is lower – since more in-domain examples are drawn as negative samples. Given these results we always use pre-ranking in the following experiments.

	without pre-ranking			with pre-ranking		
	p	r	F1	p	r	F1
Subtitles	0.722	0.984	0.833	0.964	0.978	0.971
Law	0.761	0.94	0.841	0.944	0.94	0.942
Medical	0.821	0.916	0.866	0.929	0.92	0.925
IT	0.848	0.956	0.898	0.955	0.98	0.967
Koran	0.966	0.958	0.962	0.994	0.974	0.984

Table 5: Ablation analysis showing precision (p) recall (r) and F1 for the binary classification accuracy on a held-out set, with and without pre-ranking.

4.2 Experimental Setup

We perform data selection experiments for each domain in the multi-domain dataset. As the small set of monolingual in-domain data we take the 2000 development sentences from each domain. For the general-domain corpus we concatenate the training data from all domains, resulting in 1,456,317 sentences. To enable faster experimentation we used DistilBERT (Sanh et al., 2019) for the Domain-Cosine and Domain-Finetune methods. More technical details are available in the supplementary material. We compare our methods to four approches: (1) The established method by Moore and Lewis (2010), (2) a random selection baseline, (3) an oracle which is trained on all the available in-domain data, and (4) the model we train on all the domains concatenated. We select the top 500k examples to cover the size of every specific in-domain dataset. We train Transformer NMT models on the selected data with a similar configuration to the ones trained in the cross-domain evaluation.

4.3 Results

The results are available in Table 6. We can see that all selection methods performed much better in terms of BLEU than random selection. With respect to average performance across all domains, Moore-Lewis performed better than the Domain-Cosine method, while Domain-Finetune performed best. Using the positive examples alone (Domain-Finetune-Positive) performed worse than using the top 500k examples but better than Domain-Cosine, while not requiring to determine the number of selected sentences. The average performance in

	Medical	Law	Koran	IT	Subtitles	Average
Random-500k	49.8	53.3	18.5	37.5	25.5	36.92
Moore-Lewis-Top-500k	55	58	21.4	42.7	27.3	40.88
Domain-Cosine-Top-500k	52.7	58	**22**	42.5	27.1	40.46
Domain-Finetune-Top-500k	54.8	58.8	21.8	**43.5**	27.4	**41.26**
Domain-Finetune-Positive	55.3	58.7	19.2	42.5	27	40.54
Oracle	**56.5**	**59**	15.9	43	27.3	40.34
All	53.3	57.2	20.9	42.1	**27.6**	40.22

Table 6: SacreBLEU scores for the data selection experiments. Highest scores are marked in bold.

BLEU for all data selection methods is also better than oracle selection and than training on all the available data. We perform an analysis on the selected datasets, where we measure the precision and recall of sentence selection with respect to the oracle selection. The results are available in Table 7. As also reflected in the BLEU scores, the Domain-Finetune method resulted in the highest domain recall with a minimum of 97.5, while Moore-Lewis and Domain-Cosine scored 89.4 and 78.8 respectively. We find the results very appealing given that only 2000 in-domain sentences were used for selection for each domain out of 1.45 million sentences.

	Moore-Lewis		D-Cosine		D-Finetune	
	p	r	p	r	p	r
Medical	0.476	0.955	0.391	0.788	0.485	0.975
Law	0.836	0.894	0.841	0.899	0.902	0.965
Koran	0.35	0.985	0.36	0.989	0.36	0.998
IT	0.441	0.985	0.382	0.857	0.447	0.998
Subtitles	0.899	0.899	0.916	0.916	0.957	0.957
Average	0.6	0.944	0.578	0.89	0.63	0.979

Table 7: Precision (p) and recall (r) for data selection of 500k sentences with respect to the oracle selection.

5 Related Work

Previous works used n-gram LMs for data selection (Moore and Lewis, 2010; Axelrod et al., 2011) or other count-based methods (Axelrod, 2017; Poncelas et al., 2018; Parcheta et al., 2018; Santamaría and Axelrod, 2019). While such methods work well in practice, they cannot generalize beyond the N-grams observed in the in-domain datasets, which are usually small. Duh et al. (2013) proposed to replace n-gram models with RNN-based LMs with notable improvements. However, such methods do not capture the rich sentence-level global context as in the recent self-attention-based MLMs; as we showed in the clustering experiments, autoregressive neural LMs were inferior to masked LMs in clustering the data by domain. In addition, training very large neural LMs may be prohibitive without relying on pre-training. Regarding domain clustering for MT, Hasler et al. (2014) discovers topics using LDA instead of using domain labels. Cuong et al. (2016) induce latent subdomains from the training data using a dedicated probabilistic model. Regarding vector-based data selection, Ruder and Plank (2017) learn to select data using Bayesian optimization, and explored word2vec for that purpose. Duma and Menzel (2016) create paragraph vectors for data selection in the context of SMT. Wang et al. (2017) use internal representations from the NMT model to perform data selection. Bapna and Firat (2019) propose a mechanism for incorporating retrieved sentences for each instance for domain adaptation in NMT, using representations extracted from a pre-trained NMT model. Farajian et al. (2017) explored instance-based data selection in a multi-domain scenario using information retrieval methods. Dou et al. (2019a) adapts multi-domain NMT models with domain-aware feature embeddings, which are learned via an auxiliary language modeling task. Peris et al. (2017) proposed neural-network based classifiers for data selection in SMT. For more related work on data selection and domain adaptation in the context of MT, see the surveys by Eetemadi et al. (2015) and Chu and Wang (2018). Unrelated to MT, Ma et al. (2019) used BERT to select data for tasks from the GLUE benchmark (Wang et al., 2018). However, they assumed supervision for all the different tasks/domains, while we propose an unsupervised method requiring only a small set of in-domain data.

While previous work made important contributions to domain data selection, our work is the first to explore massive pretrained language models for both unsupervised domain clustering and for data selection in NMT.

6 Conclusions and Future Work

We showed that massive pre-trained language models are highly effective in mapping data to domains in a fully-unsupervised manner using average-pooled sentence representations and GMM-based clustering. We suggest that such clusters are a more appropriate, data driven approach to domains in natural language than simplistic labels (e.g. "medical text"), and that it will improve over time as better and larger pretrained LMs will become available. We proposed new methods to harness this property for domain data selection using distance-based ranking in vector space and pretrained LM fine-tuning, requiring only a small set of in-domain data. We demonstrated the effectiveness of our methods on a new, improved data split we created for a previously studied multi-domain machine translation benchmark. Our methods perform similarly or better than an established data selection method and oracle in-domain training across all five domains in the benchmark.

This work just scratches the surface with what can be done on the subject; possible avenues for future work include extending this with multilingual selection and multilingual LMs (Conneau and Lample, 2019; Conneau et al., 2019; Wu et al.,

2019), using such selection methods with domain-curriculum training (Zhang et al., 2019; Wang et al., 2019b), applying them on noisy, web-crawled data (Junczys-Dowmunt, 2018) or for additional tasks. We hope this work will encourage more research on finding the right data for the task, towards more efficient and robust NLP.

References

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia. Association for Computational Linguistics.

Mikko Aulamo and Jörg Tiedemann. 2019. The OPUS resource repository: An open package for creating parallel corpora and machine translation services. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 389–394, Turku, Finland. Linköping University Electronic Press.

Amittai Axelrod. 2017. Cynical selection of language model training data. *arXiv preprint arXiv:1709.02279*.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. 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 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

Susan M Conrad and Douglas Biber. 2005. The frequency and use of lexical bundles in conversation and academic prose. *Lexicographica*.

Hoang Cuong, Khalil Sima'an, and Ivan Titov. 2016. Adapting to all domains at once: Rewarding domain invariance in SMT. *Transactions of the Association for Computational Linguistics*, 4:99–112.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Zi-Yi Dou, Junjie Hu, Antonios Anastasopoulos, and Graham Neubig. 2019a. Unsupervised domain adaptation for neural machine translation with domain-aware feature embeddings. *arXiv preprint arXiv:1908.10430*.

Zi-Yi Dou, Xinyi Wang, Junjie Hu, and Graham Neubig. 2019b. Domain differential adaptation for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, Hong Kong. Association for Computational Linguistics.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria. Association for Computational Linguistics.

Mirela-Stefania Duma and Wolfgang Menzel. 2016. Data selection for IT texts using paragraph vector. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 428–434, Berlin, Germany. Association for Computational Linguistics.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

M Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 152–161, Avignon, France. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423, Boulder, Colorado. Association for Computational Linguistics.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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. 2007. 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. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2019. Domain robustness in neural machine translation.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages

2814–2819, Copenhagen, Denmark. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Zuzanna Parcheta, Germán Sanchis-Trilles, and Francisco Casacuberta. 2018. Data selection for nmt using infrequent n-gram recovery. *EAMT2018*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Álvaro Peris, Mara Chinea-Ríos, and Francisco Casacuberta. 2017. Neural networks classifier for data selection in statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):283–294.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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. Association for Computational Linguistics.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2018. Data selection with feature decay algorithms using an approximated target side. *arXiv preprint arXiv:1811.03039*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Lucía Santamaría and Amittai Axelrod. 2019. Data selection with cluster-based language difference models and cynical selection. *arXiv preprint arXiv:1904.04900*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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. Association for Computational Linguistics.

Catarina Cruz Silva, Chao-Hong Liu, Alberto Poncelas, and Andy Way. 2018. Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231, Belgium, Brussels. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018.

GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566, Vancouver, Canada. Association for Computational Linguistics.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019b. Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292, Florence, Italy. Association for Computational Linguistics.

Marlies van der Wees. 2017. *What's in a Domain? Towards Fine-Grained Adaptation for Machine Translation*. Ph.D. thesis, University of Amsterdam.

Marlies van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 560–566, Beijing, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Emerging cross-lingual structure in pretrained language models. *arXiv preprint arXiv:1911.01464*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. 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 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 NMT Training

Figure 4 details the hyperparameter configuration we used to train the NMT models. We use Transformer models (Vaswani et al., 2017) in the Base configuration using the implementation provided in Fairseq (Ott et al., 2019). For all models we use a joint BPE vocabulary (Sennrich et al., 2016) learned with 32k merge operations over the concatenated corpus in both languages, enabling to tie all the embedding layers (Press and Wolf, 2017).[9] We perform early stopping if the BLEU score on the domain-specific development set did not improve in 10 consequent checkpoints. We use the ADAM (Kingma and Ba, 2014) optimizer with an initial learning rate of $5 \cdot 10^{-}4$ and a maximum of 4096 tokens per batch. We trained all models on a single NVIDIA GPU. We decode using beam search with a beam size of 5. For pre-processing we used the Moses (Koehn et al., 2007) pipeline including tokenization, normalize-punctuation, non-printing character removal, truecasing and cleaning. We removed examples with sequences longer than 100 tokens from the training data (before subword segmentation).

A.2 Data Split

Table 8 shows details about the overlap between the training, development and test sets for the different data splits of the multi-domain dataset. The overlap was computed using the English part of the corpus.

A.3 GMM Clustering

We learn GMMs with full covariance matrices, i.e. without constraints on covariance matrices that determine the shape of each component in the mixture, as implemented in scikit-learn (Pedregosa et al., 2011). We train the models until convergence or for a maximum of 150 EM iterations.

A.4 Language Model Finetuning

We fine-tune the binary classification head for 5 epochs. We use the ADAM (Kingma and Ba, 2014) optimizer with an initial learning rate of $2 \cdot 10^{-}5$. We train the model using 4 NVIDIA GPUs with 256 sentences per batch (64 per GPU).

[9] We used the implementation in https://github.com/rsennrich/subword-nmt

```
CUDA_VISIBLE_DEVICES=0 \
python $FAIRSEQ_PATH/train.py ${BINARIZED_DATA_DIR} \
  --arch transformer_wmt_en_de \
  --share-all-embeddings \
  --optimizer adam \
  --adam-betas '(0.9, 0.98)' \
  --clip-norm 1.0 \
  --lr 0.0005 \
  --lr-scheduler inverse_sqrt \
  --warmup-updates 4000 \
  --warmup-init-lr 1e-07 \
  --dropout 0.2 \
  --weight-decay 0.0 \
  --criterion label_smoothed_cross_entropy \
  --label-smoothing 0.1 \
  --max-tokens 4096 \
  --update-freq 5 \
  --attention-dropout 0.2 \
  --activation-dropout 0.2 \
  --max-epoch 200 \
  --seed 17 \
  -s $src \
  -t $tgt \
  --save-dir $MODEL_PATH \
  --save-interval-updates 10000 \
  --validate-interval 1
```

Figure 4: The hyperparameter configuration we used for NMT model training using Fairseq (Ott et al., 2019).

A.5 Moore-Lewis Implementation

We used the implementation of Moore and Lewis (2010) by Pamela Shapiro, as available in: https://github.com/pamelashapiro/moore-lewis. This implementation uses the KenLM N-Gram language model toolkit (Heafield, 2011).

A.6 Additional Visualizations

Figure 5 shows visualizations of the multi-domain dataset from additional pre-trained masked language models (BERT large and RoBERTa), and Figure 6 shows the same visualization for autoregressive models (XLNet and GPT2).

		Koehn and Knowles (2017)	Müller et al. (2019)	New Split
% dev in train	Medical	1090/2000 (54.5%)	1204/2000 (60.2%)	0/2000
	Koran	0/2000	1926/2000 (96.3)	0/2000
	Subtitles	1183/5000 (23.66%)	638/2000 (31.9%)	0/2000
	Law	595/2000 (29.75%)	1000/2000 (50%)	0/2000
	IT	2496/2526 (98.81%)	783/2000 (39.15%)	0/2000
% test in train	Medical	571/2000 (28.55%)	516/1691 (30.51%)	0/2000
	Koran	0/2000	1949/2000 (97.45%)	0/2000
	Subtitles	451/5000 (9.02%)	478/2000 (23.9%)	0/2000
	Law	649/2000 (32.45%)	966/2000 (48.3%)	0/2000
	IT	945/1856 (50.92%)	1036/2000 (51.8%)	0/2000

Table 8: Details about the different data splits for the multi-domain corpus.

Figure 5: 2D visualizations of the unsupervised GMM-based clustering for different pretrained MLMs.

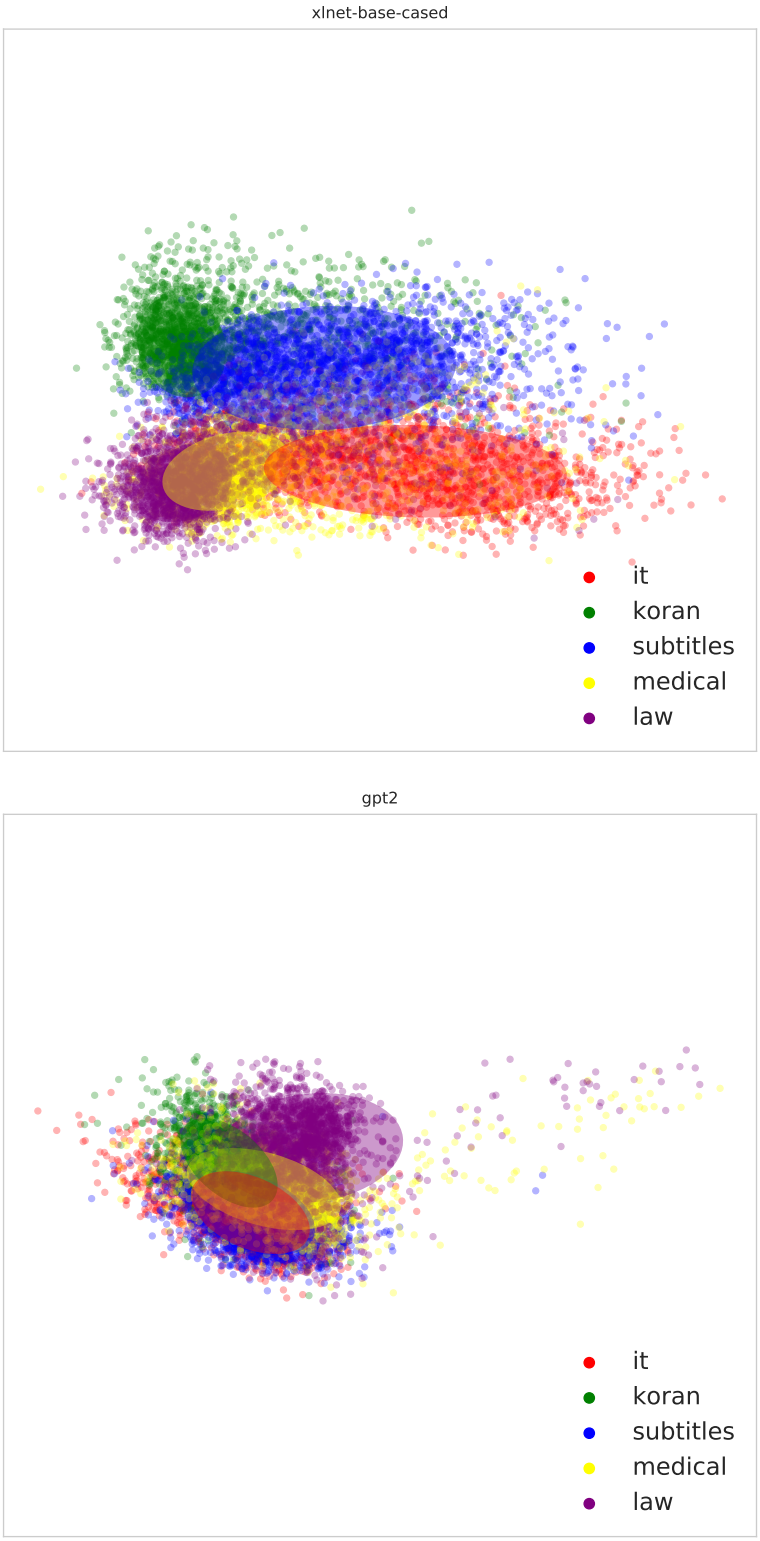


Figure 6: 2D visualizations of the unsupervised GMM-based clustering for different pretrained auto-regressive LMs.

Chapter 8

Conclusions

In this chapter I summarize the contributions described in this thesis, and present open issues and the way forward to addressing them.

8.1 Linguistically Inspired Neural Architectures

In Chapter 3, I presented neural architectures for sequence-to-sequence learning that were motivated by the monotonic relation between the characters in a word and its inflections. These "Hard-Attention" models were adopted by the community as a strong tool for morphological inflection generation (Gorman et al., 2019), lemmatization (Şahin and Gurevych, 2019), surface realization (Puzikov et al., 2019), and machine translation (Press and Smith, 2018), among others. They were also extended to non-monotonic scenarios (Wu et al., 2018a) and to exact inference (Wu and Cotterell, 2019). Our models were improved further resulting in the winning submissions for the 2017 CoNLL shared task on morphological reinflection (Makarov et al., 2017; Cotterell et al., 2017) and the 2018 CoNLL shared task on Universal Morphological Reinflection (Makarov and Clematide, 2018; Cotterell et al., 2018).

The success and proliferation of our linguistically motivated neural models for sequence-to-sequence learning encourages future research in this direction. Possible avenues for future research may include applying hard attention to

self-attention based models (Vaswani et al., 2017), which are the architectures that drive the wave of recent state-of-the-art pre-trained language models Devlin et al. (2018).

8.2 Injecting Linguistic Knowledge in Neural Models using Syntactic Linearization

In Chapter 4 I suggested to incorporate linguistic information in neural sequence-to-sequence models for machine translation by linearizing constituency trees. This approach enables to inject syntactic information to such models without changing the underlying architecture, which is very convenient given the large and growing numbers of new methods and implementations for sequence-to-sequence learning. This work and others initiated a line of work on representing syntactic trees and trees in general using neural sequence to sequence models. Some examples include dependency-based NMT (Wu et al., 2018b), syntactically supervised transformers for faster NMT (Akoury et al., 2019), and Forest-based NMT Ma et al. (2018). Regarding tasks other than MT, examples include translating between different semantic formalisms (Stanovsky and Dagan, 2018), code generation (Alon et al., 2019) and response generation Du and Black (2019).

While using syntactic information in machine translation and other NLP applications is still an active line of work, state-of-art systems in MT do not utilize such supervision (Barrault et al., 2019). Having said that, using syntax may have other appealing properties like controlling the output of text generation systems using linearized trees (Iyyer et al., 2018), generating diverse translations (Yang et al., 2019) and making translation faster Akoury et al. (2019), which makes this an exciting direction for future work.

8.3 Understanding the Weaknesses of Neural Text Generation Models

In Chapter 5 I propose better modeling and evaluation for the Split and Rephrase task (Narayan et al., 2017). The proposed improvements stemmed from an analysis of how the neural attention-based sequence-to-sequence models generate the output: I noticed that the attention mechanism consistently focused on a single entity while generating multiple output sentences, which raised suspicions regarding *how* the model learned to generate. We indeed showed that the model memorized entity-sentence pairs by introducing the model with adversarial examples which raises an important concern when using such models, especially with synthetically generated data. A consequent work by Botha et al. (2018) created a larger, more natural dataset for the task. While using our proposed modeling recipes, they greatly improved the performance on the task, showing that the original dataset was too synthetic to enable proper modeling.

To avoid such modeling deficiencies, it is important to be aware of how the dataset was created, to make sure we are modeling what we intend to model (Geva et al., 2019). These issues also call for better evaluation metrics for text generation or simplification tasks (Sulem et al., 2018), and experimental settings targeting rare words and other cases that require generalization (Shimorina and Gardent, 2018).

8.4 The Benefits of Massively Multilingual Modeling

In Chapter 6 I investigated scaling neural machine translation models to massively multilingual scenarios, involving up to 103 languages and 95.8 parallel sentences. I showed that training such models is highly effective for improving the performance on low-resource language pairs, resulting in state-of-the-art results on the publicly available TED talks dataset. I then conducted large-scale experiments pointing at the trade-off between the degradation in supervised

translation quality due to the bottleneck caused by scaling to numerous languages vs. improved generalization abilities in zero-shot translation as we increasing the number of languages.

While this work was the first to scale NMT models to such settings, many subsequent works now train massively multilingual language models that enable cross-lingual transfer learning, for better NLP in under-resourced languages (Devlin et al., 2019; Conneau et al., 2019; Siddhant et al., 2019). Other subsequent works investigate scaling such models even further in terms of the number of parameters and training examples (Arivazhagan et al., 2019) and analyzed the emerging language families within their learned representations with respect to linguistic theories on the subject (Kudugunta et al., 2019). Improving such models and making them available to the public is of very high importance and has global impact, as most of the current research on NLP is mainly focused on English (Bender, 2019).

8.5 Domain Data Selection with Massive Language Models

In Chapter 7 I show that massive pretrained language models are effective in clustering textual data to domains in a fully-unsupervised manner. I then show how to harness this property for training domain-specific machine translation models by performing domain data-selection. The proposed approach is not specific to machine translation, and can be used for any NLP task that would benefit from in-domain data.

This work is the first to exploit large pre-trained language models for domain data-selection; I would like to expand this to additional tasks and multilingual scenarios, as using proper in-domain data is crucial for building high-quality NLP systems in the real-world. While this work is still in submission while writing these lines, I expect the ideas it includes will aid future work on domain adaptation, especially in the current era where pretrained language models are one of the most common tools in the NLP practitioner's toolbox.

8.6 Going Forward

While we have witnessed great progress in the last few years regarding sequence-to-sequence learning in NLP, there are still several key subjects which I find important and that should be studied further.

8.6.1 Modeling Uncertainty

While neural sequence-to-sequence learning is very successful as a general tool for NLP tasks, this approach is hard to interpret when compared to the previously dominant phrase-based statistical machine translation methods (Koehn, 2016). In the previous methods, one could inspect the learned phrase-tables (which are essentially probabilistic dictionaries mapping words and phrases in the source language to words and phrases in the target) in order to reason about the different choices the model takes during translation. However, NMT models are composed of many millions of parameters, making it much harder to understand their inner-workings when generating an output for a given input.

This issue is very critical for improving these models further, as it is currently hard to understand or predict where and why current state-of-the-art models break. Common evaluation methods today rely on automatic metrics (e.g. BLEU (Papineni et al., 2002) or METEOR (Denkowski and Lavie, 2011)) over small human-generated test-sets which are hard and expensive to create, and may not represent the distribution of the system inputs when deployed in production settings.

It would be useful to propose methods for quantifying uncertainty in NMT models and investigate whether we can find uncertainty measures which are based solely on the model itself and unlabeled data, enabling cheap, large scale evaluation. I believe that there is enough "attack surface" to utilize for that purpose, e.g. the model parameters (and specifically the word embedding matrices), the intermediate representations the model computes, the search space the model explores during decoding and the local output distributions the model computes in each time-step.

Successfully finding such uncertainty measures will have a great impact on many down-stream applications. First of all, it will enable to detect and fix flaws in current models by getting a better understanding of what they learn, leading to improved NMT systems. It will also enable filtering of noisy corpora by finding problematic examples that cause uncertainty in the model. Another application is active learning for NMT, enabling more rapid parallel corpus creation by pointing on the most relevant examples to be professionally translated.

8.6.2 Finding the Right Data for the Task

Natural language varies greatly across topics, styles, levels of formality, genres and many other linguistic nuances (van der Wees et al., 2015; van der Wees, 2017; Niu et al., 2017). This overwhelming diversity of language makes it hard to find the right data for the task, as it is nearly impossible to well-define the exact requirements from such data with respect to all the aforementioned aspects. On top of that, domain labels are usually unavailable. Related to Section 8.6.1, a model that was never trained on data from a given domain or one that has never seen certain linguistic phenomena cannot be expected to succeed in such scenarios, and should have a high level of uncertainty in its prediction. While this observation may affect any NLP model and application, not much work is dedicated to understanding the data distribution in NLP.

Understanding how modern NLP models map data from different domains to latent neural representations can help us build more robust models, and understand why and when such models will fail given new unobserved data. While the work we proposed in Chapter 7 describes one step in this direction, there are many additional efforts that can be done on the subject, like expanding this to multilingual settings, exploring under-represented data with active learning, and proposing uncertainty measures based on the learned representations for the training data. Understanding the data landscape better can also improve unsupervised cross-lingual alignment methods which were shown to be beneficial for unsupervised neural machine translation (Conneau et al., 2019).

8.6.3 Distillation, Quantization and Retrieval for Practical Large-Scale Neural NLP

One particular reason behind the success of neural models for NLP (and specifically neural sequence-to-sequence learning) is the ability to scale such models to many millions, or even billions, of learned parameters. These large models facilitate training with very large datasets, while alleviating the parameter bottleneck (Aharoni et al., 2019). While such large models are very appealing with respect to their performance on various leader-boards, this scaling comes at the cost of increased energy consumption and latency due to the large computational costs (Schwartz et al., 2019). While larger and larger models are still being proposed (Raffel et al., 2019; Huang et al., 2019), many recent works suggest different methods to makes such models smaller and faster with relatively small losses in performance, if any (Hinton et al., 2015; Kim and Rush, 2016; Freitag et al., 2017; Tan et al., 2019; Lan et al., 2019).

I strongly believe that research work aiming to explore the model size vs. performance Pareto, as proposed in recent shared tasks on NMT efficiency (Birch et al., 2018; Hayashi et al., 2019) is very important for making progress that is both sustainable and practical for real-world use cases. In addition to distillation (Sanh et al., 2019) and quantization (Zafrir et al., 2019) efforts, future improvements may also stem from non-parametric methods (Gu et al., 2018b; Lee et al., 2019; Khandelwal et al., 2019) which relax the parameter bottleneck by adding a retrieval stage for similar examples when processing new inputs.

Chapter 9

Bibliography

Roee Aharoni and Yoav Goldberg. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1183. URL https://www.aclweb.org/anthology/P17-1183.

Roee Aharoni and Yoav Goldberg. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada, July 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-2021. URL https://www.aclweb.org/anthology/P17-2021.

Roee Aharoni and Yoav Goldberg. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2114. URL https://www.aclweb.org/anthology/P18-2114.

Roee Aharoni and Yoav Goldberg. Emerging domain clusters in pretrained language models. In *Proceedings of the 58nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Seattle, Washington, 2020. Association for Computational Linguistics. URL https://arxiv.org/abs/2004.02105.

Roee Aharoni, Yoav Goldberg, and Yonatan Belinkov. Improving sequence to sequence learning for morphological inflection generation: The BIU-MIT

systems for the SIGMORPHON 2016 shared task for morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 41–48, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2007. URL https://www.aclweb.org/anthology/W16-2007.

Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1388. URL https://www.aclweb.org/anthology/N19-1388.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1107. URL https://www.aclweb.org/anthology/N15-1107.

Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. Syntactically supervised transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1281, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1122. URL https://www.aclweb.org/anthology/P19-1122.

Uri Alon, Roy Sadaka, Omer Levy, and Eran Yahav. Structural language models for any-code generation. *arXiv preprint arXiv:1910.00577*, 2019.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D11-1033.

R Harald Baayen, Richard Piepenbrock, and Rijn van H. The {CELEX} lexical data base on {CD-ROM}. *PA: Linguistic Data Consortium, University of Pennsylvania*, 1993.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Yehoshua Bar-Hillel. The present state of research on mechanical translation. *Journal of the Association for Information Science and Technology*, 2(4):229–237, 1951.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://www.aclweb.org/anthology/W19-5301.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1209. URL https://www.aclweb.org/anthology/D17-1209.

Emily Bender. The #benderRule: On naming the languages we study and why it matters, 2019. URL https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. Findings of the second workshop on neural machine translation and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2701. URL https://www.aclweb.org/anthology/W18-2701.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino,

Lucia Specia, and Marco Turchi. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4717. URL https://www.aclweb.org/anthology/W17-4717.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6401. URL https://www.aclweb.org/anthology/W18-6401.

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/W16-2301.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1080. URL https://www.aclweb.org/anthology/D18-1080.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14, 2017.

Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1174.

R. Chandrasekar, Christine Doran, and B. Srinivas. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996. URL https://www.aclweb.org/anthology/C96-2183.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219873. URL https://www.aclweb.org/anthology/P05-1033.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007. doi: 10.1162/coli.2007.33.2.201. URL https://www.aclweb.org/anthology/J07-2003.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Do Kook Choe and Eugene Charniak. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1257. URL https://www.aclweb.org/anthology/D16-1257.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Ann Clifton and Anoop Sarkar. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1004.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2002. URL https://www.aclweb.org/anthology/W16-2002.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-2001. URL https://www.aclweb.org/anthology/K17-2001.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-3001. URL https://www.aclweb.org/anthology/K18-3001.

Jan De Belder and Marie-Francine Moens. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*. ACM, 2010.

Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2107.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805v1*, 2018.

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.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1166. URL https://www.aclweb.org/anthology/P15-1166.

Markus Dreyer and Jason Eisner. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 616–627, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D11-1057.

Markus Dreyer, Jason Smith, and Jason Eisner. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1080–1089, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1113.

Wenchao Du and Alan W Black. Top-down structurally-constrained neural response generation with lexicalized probabilistic context-free grammar. 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 3762–3771, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1377. URL https://www.aclweb.org/anthology/N19-1377.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2119.

Greg Durrett and John DeNero. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 1185–1195, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1138.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1024. URL https://www.aclweb.org/anthology/N16-1024.

Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073085. URL https://www.aclweb.org/anthology/P02-1001.

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

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Character-based decoding in tree-to-sequence attention-based neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 175–183, Osaka, Japan, December 2016a. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/W16-4617.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 823–833, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1078. URL https://www.aclweb.org/anthology/P16-1078.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2012. URL https://www.aclweb.org/anthology/P17-2012.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California, June 2016. Association for Computational Lin-

guistics. doi: 10.18653/v1/N16-1077. URL https://www.aclweb.org/anthology/N16-1077.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1101. URL https://www.aclweb.org/anthology/N16-1101.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas, November 2016b. Association for Computational Linguistics. doi: 10.18653/v1/D16-1026. URL https://www.aclweb.org/anthology/D16-1026.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674, Avignon, France, April 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E12-1068.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*, 2017.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-1035.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220296. URL https://www.aclweb.org/anthology/P06-1121.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. Factored neural machine translation. *arXiv preprint arXiv:1609.04621*, 2016.

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

Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1107. URL https://www.aclweb.org/anthology/D19-1107.

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1014. URL https://www.aclweb.org/anthology/K19-1014.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL https://www.aclweb.org/anthology/P16-1154.

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 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-1032. URL https://www.aclweb.org/anthology/N18-1032.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*, 2016.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*, 2018.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5601. URL https://www.aclweb.org/anthology/D19-5601.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

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

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, pages 103–112, 2019.

Mans Hulden, Markus Forsberg, and Malin Ahlberg. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1060. URL https://www.aclweb.org/anthology/E14-1060.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. Text simplification for reading assistance: A project note. In *Proceedings of*

the Second International Workshop on Paraphrasing, pages 9–16, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1118984.1118986. URL https://www.aclweb.org/anthology/W03-1602.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://www.aclweb.org/anthology/N18-1170.

Tomáš Jelínek. Improvements to dependency parsing using automatic simplification of data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

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://www.aclweb.org/anthology/Q17-1024.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1176.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

Katharina Kann and Hinrich Schütze. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/W16-2010. URL https://www.aclweb.org/anthology/W16-2010.

Katharina Kann and Hinrich Schütze. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the*

54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 555–560, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-2090. URL https://www.aclweb.org/anthology/P16-2090.

Ronald M. Kaplan and Martin Kay. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378, 1994. URL https://www.aclweb.org/anthology/J94-3001.

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

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://www.aclweb.org/anthology/D16-1139.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

Philipp Koehn. Computer aided translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Berlin, Germany, August 2016. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://www.aclweb.org/anthology/W17-3204.

Kimmo Koskenniemi. A general computational model for word-form recognition and production. In *Proceedings of the 4th Nordic Conference of Computational Linguistics (NODALIDA 1983)*, pages 145–154, Uppsala, Sweden, May 1984. Centrum för datorlingvistik, Uppsala University, Sweden. URL https://www.aclweb.org/anthology/W83-0114.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1565–1575, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1167. URL https://www.aclweb.org/anthology/D19-1167.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1054.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://www.aclweb.org/anthology/P19-1612.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015a.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015b. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1166.

Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Tiejun Zhao, and Eiichiro Sumita. Forest-based neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1253–1263, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1116. URL https://www.aclweb.org/anthology/P18-1116.

Peter Makarov and Simon Clematide. UZH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-3008. URL https://www.aclweb.org/anthology/K18-3008.

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-2004. URL https://www.aclweb.org/anthology/K17-2004.

Ryan McDonald and Joakim Nivre. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230, 2011. doi: 10.1162/coli_a_00039. URL https://www.aclweb.org/anthology/J11-1007.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 128–135, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-1017.

Mehryar Mohri, Fernando Pereira, and Michael Riley. A rational design for a weighted finite-state transducer library. In *International Workshop on Implementing Automata*, pages 144–158, 1997.

Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P10-2041.

Maria Nădejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Predicting target language CCG supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4707. URL https://www.aclweb.org/anthology/W17-4707.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1064. URL https://www.aclweb.org/anthology/D17-1064.

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.

Toan Q. Nguyen and David Chiang. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-2050.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1093. URL https://www.aclweb.org/anthology/N15-1093.

Xing Niu, Marianna Martindale, and Marine Carpuat. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1299. URL https://www.aclweb.org/anthology/D17-1299.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6301. URL https://www.aclweb.org/anthology/W18-6301.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 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 http://www.aclweb.org/anthology/P02-1040.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 78–85, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4009. URL https://www.aclweb.org/anthology/W14-4009.

Ofir Press and Noah A Smith. You may not need attention. *arXiv preprint arXiv:1810.13409*, 2018.

Yevgeniy Puzikov, Claire Gardent, Ido Dagan, and Iryna Gurevych. Revisiting the binary linearization technique for surface realization. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 268–278, Tokyo, Japan, October–November 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-8635. URL https://www.aclweb.org/anthology/W19-8635.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? 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 529–535, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2084. URL https://www.aclweb.org/anthology/N18-2084.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1076. URL https://www.aclweb.org/anthology/N16-1076.

Devendra Sachan and Graham Neubig. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6327. URL https://www.aclweb.org/anthology/W18-6327.

Gözde Gül Şahin and Iryna Gurevych. Two birds with one stone: Investigating invertible neural networks for inverse problems in morphology. *arXiv preprint arXiv:1912.05274*, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *arXiv preprint arXiv:1907.10597*, 2019.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2209. URL https://www.aclweb.org/anthology/W16-2209.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-3017.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in*

Natural Language Processing, pages 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1159. URL https://www.aclweb.org/anthology/D16-1159.

Anastasia Shimorina and Claire Gardent. Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 360–370, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6543. URL https://www.aclweb.org/anthology/W18-6543.

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Arivazhagan, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. *arXiv preprint arXiv:1909.00437*, 2019.

Catarina Cruz Silva, Chao-Hong Liu, Alberto Poncelas, and Andy Way. Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6323. URL https://www.aclweb.org/anthology/W18-6323.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2049. URL https://www.aclweb.org/anthology/P16-2049.

Gabriel Stanovsky and Ido Dagan. Semantics as a foreign language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2412–2421, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1263. URL https://www.aclweb.org/anthology/D18-1263.

Elior Sulem, Omri Abend, and Ari Rappoport. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1081. URL https://www.aclweb.org/anthology/D18-1081.

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., 2014a. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014b.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://www.aclweb.org/anthology/P15-1150.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*, 2019.

Masaru Tomita. Efficient parsing for natural language—a fast algorithm for practical systems. int. series in engineering and computer science, 1986.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-1059.

Marlies van der Wees. *What's in a Domain? Towards Fine-Grained Adaptation for Machine Translation*. PhD thesis, University of Amsterdam, 2017. URL https://staff.fnwi.uva.nl/m.derijke/wp-content/papercite-data/pdf/van-der-wees-phd-thesis-2017.pdf.

Marlies van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 560–566, Beijing,

China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2092. URL https://www.aclweb.org/anthology/P15-2092.

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 6000–6010, 2017.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781, 2015.

Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. Multi-perspective context aggregation for semi-supervised cloze-style reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1073.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*. ACM, 2009.

Warren Weaver. Translation. *Machine translation of languages*, 14:15–23, 1955.

P. Williams, M. Gertz, and M. Post. *Syntax-Based Statistical Machine Translation*. Morgan & Claypool publishing, 2016. ISBN 9781627059008.

Shijie Wu and Ryan Cotterell. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1148. URL https://www.aclweb.org/anthology/P19-1148.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4425–4438, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1473. URL https://www.aclweb.org/anthology/D18-1473.

Shuangzhi Wu, Dongdong Zhang, Zhirui Zhang, Nan Yang, Mu Li, and Ming Zhou. Dependency-to-dependency neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26:2132–2141, 2018b.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073079. URL https://www.aclweb.org/anthology/P01-1067.

Kenji Yamada and Kevin Knight. A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073134. URL https://www.aclweb.org/anthology/P02-1039.

Xuewen Yang, Yingru Liu, Dongliang Xie, Xin Wang, and Niranjan Balasubramanian. Latent part-of-speech sequences for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 780–790, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1072. URL https://www.aclweb.org/anthology/D19-1072.

David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Hong Kong, October 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075245. URL https://www.aclweb.org/anthology/P00-1027.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*, 2019.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016*

Conference on Empirical Methods in Natural Language Processing, pages 1568–1575, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1163. URL https://www.aclweb.org/anthology/D16-1163.

תקציר

יצירת יישומי מחשב המסוגלים לעבד בהצלחה שפה אנושית היא מטרה נכספת כבר שנים רבות בתחום הבינה המלאכותית, הנפרשת על פני משימות ושימושים רבים. דוגמא בולטת לכך היא תרגום ממוכן בין שפות שונות, משימה שמעסיקה את מוחותיהם של מדענים כבר שנים רבות (בר הלל, 1951; וויבר, 1955). בעוד שמשימה זו משמשת כמדד מדעי להתקדמות הבינה המלאכותית באופן כללי, תרגום ממוכן ומשימות עיבוד שפה טבעית (NLP) אחרות כבר נמצאות בשימוש יום־יומי בקרב מליוני אנשים ברחבי הגלובוס, ומאפשרות תקשורת טובה יותר ונגישות קלה יותר למידע בעולם.

תרגום ממוכן ומשימות עיבוד שפה טבעית נוספות ניתנות להגדרה כבעיות למידת רצף־לרצף: בעיות שמערבות קלט ופלט ששניהם רצפים של סימנים (אותיות, מילים וכו'). מנקודת מבט של למידה חישובית, בעיות כאלה כוללות חיזוי של פלט מובנה בהינתן קלט מובנה, הדורשים ייצוג עשיר של המידע ושיטות הסקה ייעודיות שלוקחות בחשבון את התלויות הרבות בין האלמנטים השונים ברצף הקלט ורצף הפלט.

ההצלחה הרבה של שיטות למידה חישובית המבוססות על רשתות עצביות (שיטות המוכרות גם בשם "למידה עמוקה") הביאה איתה גם התקדמות משמעותית בבעיות למידת רצף־לרצף. בפרט, שיטות אלה מאפשרות ללמוד ייצוגי מידע מקצה לקצה בצורה בלתי מפורשת, ללא צורך בהנדסה ידנית של מאפיינים כפי שהיה נפוץ בשיטות קודמות. בנוסף, באמצעות שיטות אלה הוסרה המגבלה של שימוש בהקשר מוגבל בקידוד כל אלמנט, מה שמאפשר מידול טוב יותר של תלויות בין אלמנטים רחוקים במשימות רצף־לרצף.

על מנת להפיק את המיטב משיטות מבוססות רשתות עצביות לבעיות עיבוד שפה טבעית, עולות שאלות מחקר רבות: כיצד עלינו לעצב מודלים עצביים תוך

כדי שילוב תובנות ידועות על שפה? איך כדאי להשתמש בייצוגים עצביים בלתי מפורשים במקרים רב־לשוניים? מהן המגבלות של שיטות אלה? מה אנו יכולים ללמוד על מידע לשוני מהייצוגים הנלמדים על ידי הרשתות העצביות?

בעבודה זו, אני מחפש תשובות לשאלות אלה ואחרות העוסקות בלמידת רצף־לרצף בעיבוד שפה טבעית. העבודה עוסקת בעיבוד שפה טבעית ברמות שונות: מורפולוגיה, התחום העוסק במחקר המילים, איך הן בנויות, ומערכת היחסים בינן לבין מילים אחרות באותה שפה, תחביר, סט החוקים, העקרונות, והתהליכים שמגדירים את מבנה המשפטים בשפה, סמנטיקה, מחקר המשמעות בשפה, בד"כ ברמת המשפט, ולבסוף פרגמטיקה, התחום העוסק בהקשר לשוני מעבר לרמת המשפט.

בפרק 3, "חילול הטיות מורפולוגיות עם תשומת־לב קשה ומונוטונית", אני מציע ארכיטקטורות עצביות חדשות ללמידת רצף־לרצף שמבטאות התאמה מונוטונית בין האלמנטים בקלט לאלמנטים בפלט באופן מפורש. המודלים שהצעתי שואבים השראה מההתאמה המונוטונית בין האותיות בהטיות המורפולוגיות השונות של מילה נתונה בשפה.

אני בוחן את המודלים שהצעתי על פני מספר מבחנים לחילול הטיות מורפולוגיות בשפות שונות, שם הם משיגים תוצאות טובות יותר ממודלים קודמים שהוצעו בספרות. הגישה שהצעתי הפכה למודל בסיס מוכר בספרות בנושא, והיא עדיין נחשבת לאחת השיטות הטובות ביותר למשימות חילול הטיות מורפולוגיות.

בפרק 4, "תרגום עצבי של מחרוזת לעץ", אני מציע שיטה לשילוב מידע לשוני במודלים עצביים לתרגום ממוכן. בשיטה זו, השואבת השראה ממודלים לניתוח תחבירי של משפטים, אני מציע לייצג את משפט היעד בתור עץ תחביר סמיכות, המקודד בתור רצף הכולל את המילים במשפט וסוגריים המבטאים את עץ התחביר.

אני מראה ששילוב ידע לשוני כזה משפר את איכות התרגום כפי שהיא נמדדת ע"י מדד BLEU וע"י שופטים אנושיים, בשפות שונות ובמיוחד במקרים בהם אין דוגמאות אימון רבות. עבודה זו היא אחד הנסיונות הראשונים לשלב מידע לשוני לתוך מודלים עצביים המאומנים מקצה לקצה, נסיונות שהובילו לקו עבודות בנושא שממשיך עד היום. עבודות המשך בחנו שילוב של מידע לשוני מסוגים נוספים, או דרכים אחרות להזרקת ידע לשוני.

בפרק 5, "פיצול ושכתוב: בסיס חזק יותר והערכת איכות טובה יותר", אני

בוחן את היכולת של מודלים עצביים לבצע משימת פישוט טקסט בה הקלט הוא משפט ארוך ומורכב והפלט הוא מספר משפטים קצרים המבטאים את אותה משמעות.

אני מראה שלמרות שמודלים עצביים מציגים תוצאות טובות יותר משנראו בעבודות קודמות על המשימה, הם רגישים להתאמת־יתר ומשננים את הפתרון במקום להכליל. לאחר מכן אני מציע חלוקה חדשה של המידע לפי המשמעות שמתארת את המשפטים השונים, מה שמאפשר לבחון את יכולת ההכללה בצורה טובה יותר ואף לחשוף את החולשות של המודלים העצביים במשימות כאלה. עבודה זו הורחבה לאוסף דוגמאות גדול ומציאותי יותר ע"י חוקרים אחרים, שאף אימצו את השיטות שהצענו.

בפרק 6, "תרגום ממוכן עצבי רב־לשוני באופן מאסיבי", אני בוחן הרחבה של תרגום ממוכן עצבי למקרים רב־לשוניים באופן מאסיבי, המערבים עד ל־103 שפות ומעל ל־95 מליון זוגות משפטים לאימון במודל עצבי אחד.

אני מראה שאימון מודלים רב־לשוניים ברמה כזו מאפשר לשפר את איכות התרגום לשפות עבורן אין דוגמאות אימון רבות, ואף מציג תוצאות טובות יותר משפורסמו קודם על תרגום אוטומטי של הרצאות TED. לאחר מכן אני עורך ניסויים בקנה מידה גדול, בהם אני מצביע על יחסי הגומלין בין הפגיעה באיכות התרגום על זוגות שפות עבורם יש דוגמאות אימון לבין השיפור באיכות התרגום על זוגות שלא היו עבורם דוגמאות אימון, ככל שמגדילים את כמות השפות המעורבות. בעוד שעבודה זו הייתה הראשונה לבצע ניסויים בקנה מידה כזה של רב־לשוניות, עבודות רבות שפורסמו מאוחר יותר מציעות אף הן מודלים רב־לשוניים המאפשרים למידת־מעבר בין שפות, לטובת עיבוד שפה טבעית לשפות ממועטות משאבים.

בפרק 7, "היווצרות אשכולות נושאיים במודלי שפה מאומנים", אני בוחן ייצוגי משפטים שנלמדו ע"י מודלי שפה שונים בקנה מידה גדול, ביחס לנושאים מתוכם נבחרו המשפטים השונים. אני מראה שבאמצעות ייצוגי המשפטים הללו ניתן לאשכל את המשפטים לפי נושא בדיוק גבוה, ובצורה בלתי מפוקחת שלא מצריכה מידע על הנושא ממנו נבחר כל משפט.

לאחר מכן אני מציע דרכים להשתמש בהיווצרות האשכולות הנושאיים לטובת בחירת מידע לאימון מודלי תרגום עצבי לנושא ספציפי. אני מראה שמודלי תרגום המאומנים בצורה כזו מגיעים לביצועים טובים יותר ממודלי בסיס חזקים

המשתמשים במידע הזמין בכל הנושאים יחדיו, או כאלה שמשתמשים בשיטות מבוססות לבחירת מידע. עבודה זו היא הראשונה לבצע שימוש במודלי שפה מאומנים בקנה מידה גדול לאשכול נושאי או לבחירת מידע בנושא, ולמעשה מציעה דרך חדשה, פרגמטית ומבוססת מידע כדי להגדיר מהם נושאים במידע טקסטואלי.

כמו שניתן לראות, למידת רצף־לרצף עם רשתות עצביות היא גישה מוצלחת להתמודדות עם מגוון בעיות ברבדים שונים של עיבוד שפה טבעית, החל ממשימות בסיס מורפולוגיות וכלה בתרגום ממוכן בין שפות רבות. למרות שגישה זו הינה שימושית במיוחד, ישנם עדיין איזורים רבים לשיפור ומחקר המשך. בחלק האחרון של עבודה זו, אני מסכם עם רשימה של נושאים למחקר עתידי בנושא שאני מוצא כחשובים במיוחד.

תוכן עניינים

תודות

לאחד והיחיד, יואב גולדברג, המנחה שלי במסע שמסתכם בעבודה זו. תודה על כל הסבלנות, ההשראה, הידע האינסופי, היצירתיות, ובאופן כללי כוחות העל שלך. בזכותך המסע הזה היה תענוג, והשנים האחרונות היו מהמהנות והמלמדות ביותר בחיי. חוץ מעל עיבוד שפה טבעית, למדתי ממך כל כך הרבה על איך לעשות מדע כמו שצריך, איך לשאול את השאלות הנכונות ואיך לתקשר רעיונות מורכבים בצורה ברורה ועניינית. לא יכלתי לבקש מנחה טוב יותר, ואני מעריך מאוד את כל מה שלמדתי ממך. תודה!

לכל חברי המעבדה לעיבוד שפה טבעית בבר אילן, ובמיוחד לעידו דגן, אלי קיפרווסר, ורד שוורץ, גבי סטנובסקי, ינאי אלעזר ועמית מור־יוסף, תודה על שהייתם קולגות מדהימים, ועל שיצרתם סביבה תומכת ומקבלת שהפכה את המחקר לחוויה מדהימה. למדתי כל כך הרבה מכם ואני בטוח שנשתף פעולה בעתיד.

לאורהן פיראט, מלווין ג'ונסון ושאר חברי צוות גוגל טרנסלייט, תודה לכם על שנתתם לי את ההזדמנות לעבוד איתכם באחד מצוותי עיבוד השפה הטבעית המשפיעים ביותר בעולם. קיץ 2018 היה בלתי נשכח וסוג של הגשמת חלום, ואין ספק שהוא עיצב את עתידי המקצועי בצורה משמעותית.

אחרונים חביבים, ארצה להודות למורים שעזרו לי ללמוד ולהעמיק בתרגול היוגה. התרגול תרם רבות להיותי חזק וגמיש, גם מבחינה פיסית, אך בעיקר מבחינה מנטלית במהלך המסע הזה. נמסטה.

להוריי, שושנה (שושי) ויהושע (שוקי). תודה על כל האהבה האינסופית, התמיכה, ועל שגידלתם אותי להיות סקרן תמיד. לספיר, החברה הכי טובה שלי והשותפה שלי בחיים. תודה על האהבה והתמיכה למרות סופי השבוע העסוקים, הנסיעות הארוכות והלילות הלבנים לפני דדליינים.

העבודה הזו מוקדשת לכם.

עבודה זו בוצעה בהנחייתו של פרופ' יואב גולדברג, המחלקה למדעי המחשב, אוניברסיטת בר־אילן.

נושאים בלמידת רצף לרצף
בעיבוד שפה טבעית

חיבור לשם קבלת התואר "דוקטור לפילוסופיה"

מאת:

רועי אהרוני

הוגש לסנט של אוניברסיטת בר־אילן

רמת גן　　אייר תש"פ