UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Injection of linguistic knowledge into neural text generation models

Noe Casas

Universitat Politècnica de Catalunya

PhD Thesis

Injection of Linguistic Knowledge into Neural Text Generation Models

Noe Casas

Advised by:

Marta R. Costa-jussà

José A. R. Fonollosa

October, 2020

Abstract

Language is an organic construct. It emanates from the need for communication and changes through time, influenced by multiple factors. The resulting language structures are a mix of regular syntactic and morphological constructions together with divergent irregular elements. Linguistics aims at formalizing these structures, providing a rationalization of the underlying phenomena. However, linguistic information alone is not enough to fully characterize the structures in language, as they are intrinsically tied to meaning, which constrains and modulates the applicability of the linguistic phenomena and also to context and domain.

Classical machine translation approaches, like rule-based systems, relied completely on the linguistic formalisms. Hundreds of morphological and grammatical rules were wired together to analyze input text and translate it into the target language, trying to take into account the semantic load carried by it. While this kind of processing can satisfactorily address most of the low-level language structures, many of the meaning-dependent structures failed to be analyzed correctly.

On the other hand, the dominant neural language processing systems are trained from raw textual data, handling it as a sequence of discrete tokens. These discrete tokens are normally defined looking for reusable word pieces identified statistically from data. In the whole training process, there is no explicit notion of linguistic knowledge: no morphemes, no morphological information, no relationships among words, or hierarchical groupings.

This thesis aims at bridging the gap between the neural systems and linguistics-based systems, devising systems that have the flexibility and good results of the former with a base on the linguistic formalisms, with the purposes of improving quality where data alone cannot and forcing human-understandable working dynamics into the otherwise black-box neural systems. For this, we propose techniques to fuse statistical subwords with word-level linguistic information, to remove subwords altogether and rely solely on lemmas and morphological traits of the words, and to drive the text generation process on the ordering defined by syntactic dependencies.

The main results of the proposed methods are the improvements in translation quality that can be obtained by injecting morphological information into NMT systems when testing on out-of-domain data for morphologically-rich languages, and the control over the generated text that can be gained by means of linking the generation order to the syntactic structure.

Acknowledgements

Research is not easy, especially when you are starting in a field that is new to you, like my case. Apart from this inherent difficulty of research itself, there are also several obstacles and hardships a PhD candidate has to go through. If I have been able to walk this path, it is not only due to my personal effort but, crucially, due to the support of many people. In these few lines, I want to thank them.

Thanks to my advisors, Marta and Adrián, who believed in me, helped me find funding, passed their knowledge onto me, and guided my research. This journey would not have been possible at all without them.

Thanks to Lucy Software, especially to Juan Alonso, who opened the industrial PhD position that unlocked my dream of pursuing research. Thanks for his encouragement and for the flexibility he has gifted me. And thanks to the rest of my colleagues at Lucy, who have created such a welcoming work environment.

Thanks to my lab mates, especially to Carlos, Casimiro, Magdalena, Christine and Bardia. It has been great to endure this path by your side, sharing experiences, and encouraging each other to push forward. I will always cherish our trip to ACL 2019 in Florence.

Thanks also to Professor Xavier Giró and all those that made it possible for me to teach at UPC. I love teaching.

Thanks to the Catalan Agency for Management of University and Research Grants (AGAUR), which funded the Industrial PhD Grant that made it possible for me to attend conferences where I learned so much. Also, to the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund, and the Spanish Agencia Estatal de Investigación, for supporting my advisor, Marta, through the postdoctoral senior grant Ramón y Cajal, and through the projects TEC2015-69266-P (MINECO/FEDER,EU), EUR2019-103819 and PCIN-2017-079.

And finally, the most important. Thanks to my mother, Carmen, and my girlfriend, Laura, who have supported me in every stage of the PhD, as they always do at all moments in life. They encouraged me to pursue my idea of starting a PhD at 35. They patiently listened to my explanations about my progress. They cheered me up when I needed it. They celebrated my every small achievement. I love you.

Contents

1. Introduction **1**
1.1. Thesis Contributions 2
1.2. Publications 3
1.3. Thesis Structure 5

2. Background and Related Work **7**
2.1. Natural Language Processing 7
2.2. Machine Translation Paradigms 8
2.3. Neural Netwoks for Text Generation 9
2.4. Linguistic Knowledge and Where to Find It 24
2.5. Linguistic Knowledge in Neural Models 29

3. Linguistic Knowledge-based Vocabularies for Neural Machine Translation **35**
3.1. Morphological Unit Vocabulary 36
3.2. Lemmatized Vocabulary 38
3.3. Experiments 41
3.4. Discussion 48
3.5. Conclusion 51

4. Sparse factored Neural Machine Translation **53**
4.1. Sparse factored NMT 54
4.2. Experimental Setup 55
4.3. Results 58
4.4. Conclusion 59

5. Combining Subword Representations into Word-level Representations **61**
5.1. Subword to Word Transformer 62
5.2. Experimental Setup 64
5.3. Results 65
5.4. Conclusion 67

6. Syntax-driven Iterative Expansion Language Models for Text Generation **69**
6.1. Iterative Expansion LMs 69
6.2. Experimental Setup 73
6.3. Datasets and Preprocessing 74
6.4. Hyperparameter Configuration 75
6.5. Results and Analysis 77
6.6. Further Comparison with Real Text 80
6.7. A Note on Perplexity Computation 86
6.8. Conclusion 86

7. Conclusion **87**

Bibliography **89**

Appendix A. International MT Evaluation Campaigns **109**
A.1. Third Conference on Machine Translation (WMT18) 109
A.2. Fourth Conference on Machine Translation (WMT19) 116

1. Introduction

Machine translation (MT) is the area within natural language processing (NLP) that devises systems to translate text from a source language into a target language. Since the early days of computation, different approaches have been applied to try to address translation tasks. Currently, the dominant paradigm is neural machine translation (NMT), which relies on artificial neural networks.

While the quality of the translations generated by NMT systems is higher that the preceding MT paradigms, they are not error-free. Some of the errors present in NMT translations include repeated words, inability to handle words that were not part of the training data, addition of extra information that was not present in the source sentence or missing parts from the original source sentence information.

Neural networks are said to be "black boxes", as we cannot understand the behaviour of the function learned by the network during training. This way, when the network generates an undesirable output, we cannot identify any reason why the output is different from the desirable one. This makes neural networks difficult or impossible to fix in a traditional way a software bug is fixed. In the same line, translations generated by NMT systems are not interpretable, in the sense that we cannot establish a causal relationship between the parts of the input sentence, specific pars of the computation and the generated output.

NMT systems are trained on raw textual data. This training data consists of a collection of pairs of source sentence and its translation in the target language. The number of sentence pairs in a normal training dataset is in the range of millions. These sentences are segmented into smaller pieces of text called tokens, normally defined at the level of words or even subword segments, and then the most frequent tokens are selected to be part of the finite set of elements that the MT system will handle, the vocabulary.

The ubiquitous vocabulary definition strategy used currently in NMT systems is called byte-pair encoding (BPE). It finds subword segments that are statistically highly reusable in the training data. These subwords, nevertheless, have not morphological grounding and may be totally unrelated to the morphological segmentation that a linguist may apply to words, preventing any morphological interpretation of the internal model dynamics.

For the neural network, each sentence or document is just a sequence of symbols from the vocabulary. The neural network, therefore, does not receive any information about the syntax of the sentence or relationships among words or groups of words, or the function each word plays in the sentence structure.

The fact that the information handled by the neural network is totally unrelated to morphology, syntax or any other linguistic framework, together with the black-box nature of neural networks, makes NMT translations unexplainable, even more if compared with other formerly dominant MT paradigms, like rule-based machine translation (RBMT), that relied completely on linguistic information and were fully interpretable. This problem is more severe when NMT systems are used to translate texts that are of a different domain that the ones used to train it, as in these cases where the translations tend to have high fluency but low adequacy (Koehn and Knowles, 2017).

In this regard, some lines of research have tried to "inject" linguistic information into NMT systems, either with the purpose of improving the end results or to make neural network internals closer to concepts that humans can understand better.

The works developed as part of this thesis go in that line, aiming at bridging the benefits that linguistic information brought to RBMT systems with the performance of the currently dominant NMT systems.

1.1. Thesis Contributions

The goal of this thesis is to study different approaches to profit from linguistic knowledge in neural text generation models. Specifically, the contributions to be found in this thesis are:

- Linguistic Knowledge-based Vocabularies for Neural Machine Translation: two linguistically-grounded approaches to extract the vocabulary of NMT systems are proposed, studying their benefits in comparison with the dominant subword-based NMT models, under different data scenarios.
- Sparse factored Neural Machine Translation: a novel approach to inject linguistic knowledge into NMT where the linguistic annotation scheme is not dense but sparse.
- Combining Subword Representations into Word-level Representations: a reformulation of the Transformer NMT architecture is proposed, aiming at

combining the subword token representations into word-level representations and providing a natural point to incorporate extra word-level linguistic or semantic information.

- Syntax-driven Iterative Expansion Language Models for Text Generation: a new paradigm for introducing a syntactic inductive bias into neural text generation, where the dependency parse tree is used to drive the Transformer model to generate sentences iteratively.

1.2. Publications

This dissertation presents several contributions to the incorporation of linguistic information into neural text generation systems. Publications that are a direct result of this work include:

- N. Casas , M. R. Costa-jussà, J. A. R. Fonollosa, J. A. Alonso, and R. Fanlo. Linguistic knowledge-based vocabularies for neural machine translation. *Natural Language Engineering*, page 1–22, 2020c. URL https://doi.org/10.1017/S1351324920000364

- N. Casas , M. R. Costa-jussà, and J. A. R. Fonollosa. Sparsely factored neural machine translation. 2020a. Under review

- N. Casas , M. R. Costa-jussà, and J. A. R. Fonollosa. Combining subword representations into word-level representations in the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 66–71, Online, July 2020b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-srw.10

- N. Casas , J. A. R. Fonollosa, and M. R. Costa-jussà. Syntax-driven iterative expansion language models for controllable text generation. 2020d. Accepted for publication at the EMNLP 2020 Workshop on Structured Prediction for NLP

During the course of this thesis, other side works were developed, leading to publications. While they are not specifically focused on the injection of linguistic information, they are about relevant topics related to NMT. Among them, the ones led by the author of this thesis were:

- N. Casas , J. A. Fonollosa, and M. R. Costa-jussà. A differentiable BLEU loss. Analysis and first results. Presented at the Workshop of the International Conference on Learning Representations (ICLR), 2018b. URL https://openreview.net/forum?id=HkG7hzyvf
- N. Casas , C. Escolano, M. R. Costa-jussà, and J. A. R. Fonollosa. The TALP-UPC machine translation systems for WMT18 news shared translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 355–360, Belgium, Brussels, Oct. 2018a. Association for Computational Linguistics. doi: 10.18653/v1/W18-6406. URL https://www.aclweb.org/anthology/W18-6406
- N. Casas , J. A. R. Fonollosa, C. Escolano, C. Basta, and M. R. Costa-jussà. The TALP-UPC machine translation systems for WMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5311. URL https://www.aclweb.org/anthology/W19-5311

The collaborations were the author of this thesis was not the main author are the following:

- D. Torregrosa, N. Pasricha, M. Masoud, B. R. Chakravarthi, J. Alonso, N. Casas , and M. Arcan. Leveraging rule-based machine translation knowledge for under-resourced neural machine translation models. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 125–133, Dublin, Ireland, Aug. 2019. European Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6725
- C. Basta, M. R. Costa-jussà, and N. Casas . Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39. Association for Computational Linguistics, Florence, Italy, Aug. 2019. doi: 10.18653/v1/W19-3805. URL https://www.aclweb.org/anthology/W19-3805
- C. Basta, M. R. Costa-jussà, and N. Casas . Extensive study on the underlying gender bias in contextualized word embeddings. *Neural Computing and Applications*, 2020. URL https://doi.org/10.1007/s00521-020-05211-z
- M. Artetxe, G. Labaka, N. Casas , and E. Agirre. Do all roads lead to Rome? Understanding the role of initialization in iterative back-translation.

Knowledge-Based Systems, page 106401, 2020. ISSN 0950-7051. doi: 10.1016/j.knosys.2020.106401. URL http://www.sciencedirect.com/science/article/pii/S0950705120305335

- M. R. Costa-Jussà, N. Casas , C. Escolano, and J. A. R. Fonollosa. Chinese-Catalan: A neural machine translation approach based on pivoting and attention mechanisms. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4):1–8, 2019. URL https://dl.acm.org/doi/abs/10.1145/3312575

- M. R. Costa-jussà, N. Casas , and J. A. Fonollosa. English-Catalan neural machine translation in the biomedical domain through the cascade approach. In *Proceedings of the Multilingual Biomedical Text Processing Workshop of the 11th Language Resources and Evaluation Conference of the European Language Resources Association*, 2018. URL http://temu.bsc.es/multilingualbio2018/wp-content/uploads/2018/03/LREC-2018-PROCEEDINGS-MultilingualBIO.pdf

- J. A. R. Fonollosa, N. Casas , and M. R. Costa-jussà. Joint source-target self attention with locality constraints. 2019. URL https://arxiv.org/abs/1905.06596. Under review

1.3. Thesis Structure

This thesis presents and summarizes the work from some of the publications listed in Section 1.2, reusing part of them and tailoring it as appropriate.

Chapter 2 provides the needed background to the rest of the thesis, presenting as well the state of the art of the relevant subareas each contribution belongs to. Each subsequent chapter cites the appropriate sections from the background chapter that are relevant to it.

Chapter 3 studies novel morphologically-grounded strategies to define the vocabulary of the source-side of NMT systems (Casas et al., 2020c).

Chapter 4 also addresses strategies for incorporating morphological knowledge as input to NMT systems, but focuses on linguistic annotation schemes that are not dense but sparse in terms of annotated features (Casas et al., 2020a).

In Chapter 5 we reformulate the Transformer model to better incorporate word-level linguistic information in subword-based NMT architectures (Casas et al., 2020b).

In Chapter 6, a new text generation paradigm driven by the sentence syntactic structure is presented (Casas et al., 2020d).

Finally, Chapter 7 presents the conclusions drawn from this work.

Additionally, Appendix A describes the participation in international MT evaluation campaigns (Casas et al., 2018a, 2019).

2. Background and Related Work

In this chapter we provide an overview of the topics that are most relevant to the work presented in this thesis, regarding both foundational concepts as well as state of the art research.

In section 2.1 we provide a brief overview of natural language processing (NLP) in general, describing the NLP tasks that relate most to the ones presented in this work. In section 2.2 we provide some background on machine translation and the different paradigms that have dominated the MT paradigm over the years. In section 2.3 we describe neural MT and neural language models. In section 2.4, we describe what kind of linguistic knowledge is used in the work presented in this thesis, and identify the sources to extract it. Finally, in section 2.5 we describe how such kinds of linguistic knowledge have been imbued into neural models in the literature.

2.1. Natural Language Processing

Natural Language Processing is the area that studies the processing of natural language textual data. This area is very broad, and comprises several tasks where different problems are explored. In this section we describe some of the relevant ones in relation with the use of linguistic knowledge and with the work in this thesis.

Sentence classification aims at assigning to sentences a label that marks they belong to a specific category within a taxonomy. An example is sentiment classification, where the sentence can labeled as positive, negative or neutral. A special case of sentence classification is natural language inference (NLI), sometimes referred to as textual entailment, which receives two sentences and classifies their relationship as entailing, contradictory or neutral. The architectures for sentence classification are very related with the encoder part of NMT models and, many of the approaches used to incorporate linguistic knowledge for text classification have also been applied for NMT.

Word Annotation tasks try to assign a tag to each word in the text. POS tagging is the most representative word annotation task, together with lemmatization.

Syntactic structure analysis tasks annotate the words and their relationships, like in constituency parsing, dependency parsing. Given their relationship with linguistic knowledge, POS tagging, constituency parsing and dependency parsing are further described in section 2.4.

Language Modeling is the task of modeling the probability distribution of text, that is, estimating how probable a sentence or a piece of text is in the given language. This task is further explored in section 2.3.3.

Machine Translation is the task of receiving a sentence in a source language and translating it into a target language. This task is further explored in section 2.3.2.

2.2. Machine Translation Paradigms

The first machine translation systems were **Dictionary-based**, using bilingual dictionaries to translate word by word the source sentence into the target language. Such an approach neglected any notion of syntax or context-dependent meaning and offered poor results. This led to the appearance of **Rule-based Machine Translation (RBMT)** systems, which made use of formal grammars to analyze the source sentences and to transform them into the target language. While the results obtained by these systems were far better than with dictionary-based systems, their translations frequently lacked idiomatic constructions and were perceived as mechanical. While RBMT systems are not the focus of this thesis, they are used as a source of linguistic knowledge to imbue into neural systems; being relevant to this work, they are further described in section 2.4.4

Statistical Machine Translation (SMT) (Brown et al., 1993; Koehn et al., 2003) systems tried to mitigate the lack of idiomatic translations by using *translation examples* as a means to *learn* how to translate from source to target language. SMT systems need therefore to be *trained* before they can actually be used for translation (i.e. inference). The input of the SMT training is a large parallel corpus, that is, a collection of translation example pairs, each one containing a sentence in the source language and its translation in the target language. The size of parallel corpora used for SMT ranged from several hundred thousand parallel sentences to several million parallel sentences. SMT training is based on devising an alignment model that finds out the correspondence between words in the source sentence and words in the associated target sentence. This model is subsequently used to compute a probability table, which contains a mapping between words in the source language and their equivalent translations in the target language, together with the estimated probability of such a translation,

and the same for the opposite translation direction. This type of word-level probability table was later evolved to contain small sequences of words (i.e. phrases) instead of individual words, and were called phase tables. The third element in a SMT system is a language model (LM) of the target language, a statistical model that can evaluate the likelihood of an arbitrary word sequence to be a correct utterance in the target language. When the SMT system is used for inference, these three models are leveraged to first compute the most probable target phrase correspondences and their ordering and then to score them to draw the one showing the highest probability of being the correct translation of the source probable sentence.

SMT ceased to be the dominant MT paradigm in the second half of the 2010 decade. Neural Machine Translation (NMT) models, which also rely on large parallel corpora to be trained, are currently the state of the art paradigm for MT, having translations perceived as more idiomatic and natural compared to SMT. NMT is further described in section 2.3.2.

2.3. Neural Netwoks for Text Generation

Neural networks regained attention after the success of Krizhevsky et al. (2012) with the AlexNet convolutional architecture for image classification and the appearance of GPU hardware capable of powering the neural computations at scale in affordable time.

Later, neural networks achieved success also in NLP, first in discriminative tasks (e.g. classification) and later in generative tasks (e.g. translation). Some early factors that constrained the application of neural networks to NLP tasks were the discrete nature of text (as opposed to the continuous signals from the image domain) and the variable length of textual sequences.

The mapping of discrete textual tokens to continuous representations made it possible to represent text in an appropriate format for the inputs of neural networks. However, given the multiple levels of granularity at which information is encoded in text, together with the inherent limitations of the current neural models, text representation is not yet totally solved. This aspect is further elaborated in section 2.3.1.

The variable length of textual sequences was initially mitigated in discriminative models with recurrent architectures, like LSTMs, that allowed to accumulate a joint representation of the whole sequence. In generative models, recurrent architectures used in an autoregressive manner were also the initial response, using a conditioning signal when needed, like in NMT. Attention mechanisms

first were used to complements recurrent units and later to replace them as main computational building block. In sections 2.3.3 and 2.3.2 we describe in detail the dominant neural architectures for LM and NMT in the last years, while in section 2.3.4 we explore the recent lines of research of non-autoregressive text generation.

2.3.1. Representation of Textual Data

In NLP tasks, text is received either as input or generated as output (e.g. machine translation, language modeling). In order to process text, it is common for neural networks applied to NLP tasks to split the original character string into a sequence of substrings, and to represent each substring as a discrete token. The granularity used to split the original text into substrings is part of the design of any NLP system.

Languages themselves offer information packaged at different natural granularity levels: sub-character information (e.g. radicals in Chinese characters), characters, morphemes, words, multi-word expressions, sentences and documents. Apart from the linguistically natural information packages, it is also possible to build synthetic partitions (e.g. statistically-discovered subwords (Sennrich et al., 2016c), byte-level representations (Costa-jussà et al., 2017)) as well as hybrid granularity levels (e.g. hybrid word-character representations (Luong and Manning, 2016)).

The representation granularity defines how to split a piece of text into a sequence of discrete tokens and is a key design aspect in any NLP system because it determines the type of information it can directly profit from. This way, a word-level system can profit from word-level information (e.g. semantics), while a character-level system does not have direct access to such a type of information.

The set of all possible tokens is referred to as vocabulary and, normally, the higher the representation granularity, the larger the size of the vocabulary. This way, the set of all possible words is larger than the set of all possible characters. Nevertheless, given the open nature of language, any finite size word-level vocabulary is to face the problem of words that are not part of the vocabulary and hence can not be properly represented.

The selection of an appropriate granularity level is also influenced by the capability of the downstream NLP system to handle the resulting vocabulary. This way, while symbolic systems can handle very large vocabularies (i.e. several hundred thousand different tokens), current neural networks can only handle moderately large vocabularies (i.e. tens of thousand different tokens). This makes is desirable

for Neural network-based NLP systems to keep the vocabulary size constrained while trying to maximize the representation ability.

The vocabulary is defined prior to the training of the neural network, normally by means of an algorithmic approach that "extracts" the possible tokens from the training data according to the chosen token granularity.

Character-level vocabularies define one token for each different character present in the training data. Their size ranges from tens to thousands of characters, depending on the language. In English, this would include all letters, both lowercase and uppercase, punctuation symbols, blanks, etc. A character-level vocabulary allows to represent any text that contains the characters in the vocabulary, not only the words from the training data.

Word-level vocabularies define a token for each different word present in the training data. Given the huge amount of different words, only the N most frequent words are kept in the vocabulary, dropping the less frequent ones. The selection of hyperparameter N is driven by different factors, including hardware memory constraints, scaling limitations of the network architecture (e.g. softmax for network output) and the scarceness of lower frequency words in the training data (it is not useful to represent words whose frequency of appearance in the training data is not enough for the network to learn how to use them). A frequent default value is $N = 32k$ tokens. A special token `<UNK>` is usually introduced in the vocabuly in order to represent words that are not part of the vocabulary (i.e. unknown words, or out-of-vocabulary (OOV) words).

Multi-word level vocabularies extend word-based ones and try to find sequences of words that conform a single lexical unit or are part of an idiomatic construct (Mikolov et al., 2013).

Subword vocabularies have word-pieces as tokens, which are extracted statistically from the training data based on their frequency of appearance. For languages with regular morphology, extracted subwords may match morphological word parts, however, there is no guarantee of morphological soundness. Subword vocabularies normally do not have an `<UNK>` token because, apart from the multi-character subwords, there are usually single-character subwords that allow to represent any input text. The currently dominant subword vocabulary extraction approach is **Byte-Pair Encoding (BPE)** (Sennrich et al., 2016c). This approach consists in taking all words from the training data and building subwords starting from a character-based vocabulary (with all characters present in the training data) and creating new tokens by iteratively merging the two tokens that appear together most frequently. Subwords that can be followed by other subwords are normally marked with suffix `@@`, which is removed when decoding text back. This is illustrated in Figure 2.1. BPE and some of its variants, such

as word-pieces (Wu et al., 2016) are the dominant subword vocabulary definition strategy in the state of the art neural machine translation (NMT) architectures.

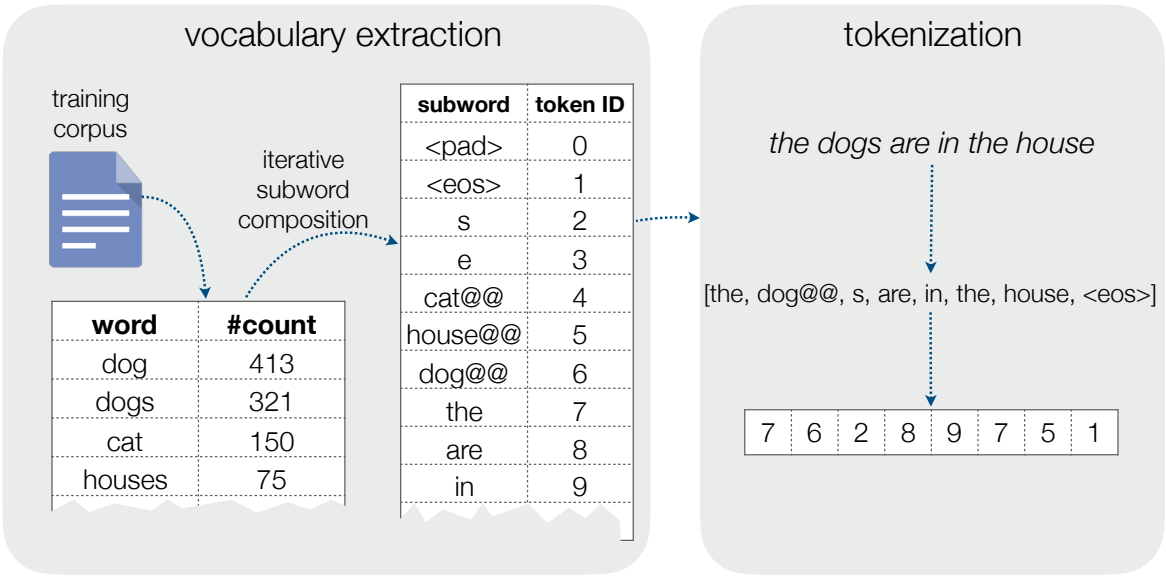

Figure 2.1.: Byte-Pair Encoding (BPE) vocabulary extraction and tokenization.

Despite the flexibility of character-level vocabularies, they delegate the learning of word formation to the network and the resulting token sequences are very long, which, for some tasks like machine translation (MT), leads to a decrease in the quality (Gao et al., 2020). On the other hand, word-level vocabularies relieve the network completely from learning word formation, but they frequently lead to OOV words and they aren't aware of the connection of different forms of the same word, leading to worse training data utilisation, especially for highly inflected languages and agglutinative languages. Subword vocabularies are a compromise between both, and are indeed used in the current state of the art of several NLP tasks, like MT.

Nevertheless, the benefits of word-level vocabularies lie in the fact that tokens can be associated with the word they represent, which can be key to certain tasks related to the meaning of the word or setups related to the word-level granularity (reuse of pretrained word embeddings for sentiment classification, induction of cross-lingual word embeddings); character and subword vocabularies lack such a trait and this makes them less suitable for such tasks.

There have been attempts to profit from word-level information in subword-based vocabularies. These approaches addressed in different ways the mismatch between those two different token granularity levels. The approaches by Bojanowski et al. (2017), Zhao et al. (2018) and Li et al. (2018) aim at computing

pre-trained word representations from the subword information. Other proposals integrate the computation of the word representation in the overall neural model, either combining information from character level, like those by Luong and Manning (2016) Costa-jussà and Fonollosa (2016), from n-gram level, like the one by Ataman and Federico (2018), or from multiple granularities like the work by Chen et al. (2018). Some other approaches like those by Wang et al. (2019) and Gu et al. (2018b) try to extend this idea to obtain multilingual *conceptual* representations from character-level representations.

2.3.2. Neural Machine Translation

Neural Machine translation (NMT) networks model the probability of each of the tokens in the translation y_t conditioning on both the source sequence tokens $x_1, \ldots, x_T$ and also on the previous tokens $y_{<t}$:

$$p\left(y_t | x_1, \ldots, x_T, y_1, \ldots, y_{t-1}\right) \tag{2.1}$$

Given (2.1), we can formalize the probability of the whole translation using an autoregressive factorization as shown in (2.2).

$$p\left(y_1, \ldots, y_{T'} | x_1, \ldots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | x, y_1, \ldots, y_{t-1}\right) \tag{2.2}$$

The inputs to the neural network are therefore the source sentence tokens $x_1, \ldots, x_T$ and the previous tokens from the target sentence already generated by the network $y_1, \ldots, y_{t-1}$. The output of the network is a probability distribution over the target token space, from which a token is selected at each time step of the generation process, so that the translation is generated token by token.

NMT is the currently dominant MT paradigm, showing a good level of translation quality for high-resource language pairs. In the following sections we explore the different aspects that characterize NMT systems, including their neural architectures that have been dominant over time, the different approaches to draw tokens from the probability distribution over the token space (i.e. decoding) and the evaluation measures used to gauge the quality of the translations generated by the model.

Neural Architectures

The first successful NMT models were **sequence to sequence architectures** (Sutskever et al., 2014), which consist in an encoder-decoder structure where both the encoder and decoder are recurrent cells (either vanilla recurrent neural networks (RNN), long-short term memories (LSTM; Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRU; Cho et al., 2014)).

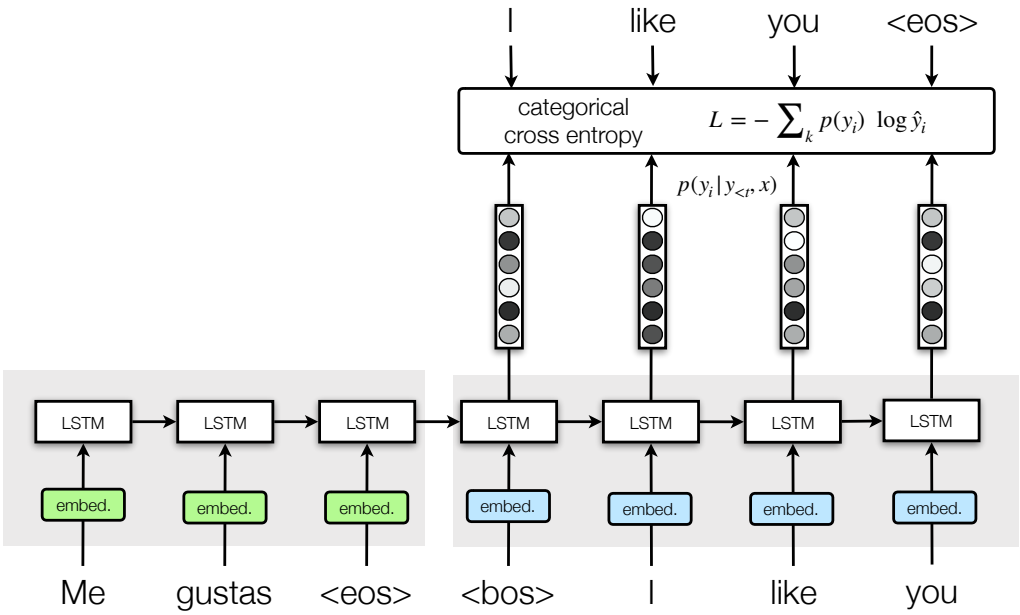

Figure 2.2.: Sequence to sequence architecture at training time.

The recurrent units at encoder and decoder are unrolled at training time, using a back-propagation through time scheme. As shown in figure 2.2, in the encoder part:

- the input sequence tokens are embedded and fed to the encoder units,
- the context vector is passed from one unit to the next one,
- the output of all encoder units is discarded,

while in the decoder part:

- the first unit receives as input context vector the output context vector of the last encoder unit,
- for each unit, the token generated by the previous unit is received as input, except for the first unit, which receives the input BOS (beginning of sentence) token,
- the output is a softmax representing the categorical probability distribution over the output token space.

For decoder unit at position i, the expected output is the token at position i in the target sequence. The expected tokens are normally represented as one-hot vectors, or their label smoothed version (Szegedy et al., 2016). The loss function used as minimization objectively is defined as the categorical cross entropy between the output of the model and the expected output, taking into account only the tokens that appear before the EOS token in the expected output.

It is frequent to use a variation of the back-propagation through time; this variation is called teacher forcing (Williams and Zipser, 1989): at training time, the gold data tokens are used as outputs of the decoder units instead the prediction of the previous unit. This variation allows to train in parallel (as one unit does not need the output of the previous one). Nevertheless, as at inference time the model is still used in an autoregressive manner, using teacher forcing induces an exposure bias in the network as it is only trained with the real data as prefix, but at inference time it uses its own predictions instead.

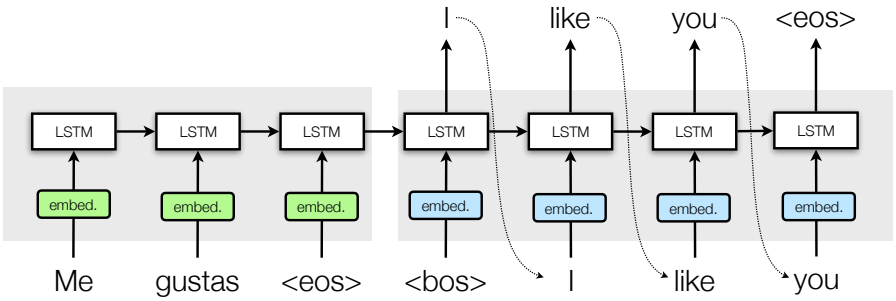

Figure 2.3.: Sequence to sequence architecture at inference time.

At inference time, this model is used in an **autoregressive** way, first predicting token at the first position, then feeding such token to the input of the decoder and computing the output again, and so on, as shown in Figure 2.3. This autoregressive nature makes the computation of the translation linear on the output sequence length.

A notable problem of sequence to sequence models is that the encoder has to fit all the information from the source sentence into a fixed length vector representation (i.e. the context vector passed from encoder to decoder). This posed an information bottleneck that was overcome by the introduction of an **attention mechanism** (Bahdanau et al., 2015; Luong et al., 2015) that allowed the decoder to use a weighted sum of all context vectors from the encoder. The results of sequence to sequence with attention outperformed those of vanilla sequence to sequence models. This is illustrated in Figure 2.4.

After sequence-to-sequence models, Convolutional NMT (Gehring et al., 2017) showed promising results, but it was soon surpassed by the current state of the art NMT architecture, the **Transformer** model (Vaswani et al., 2017). This

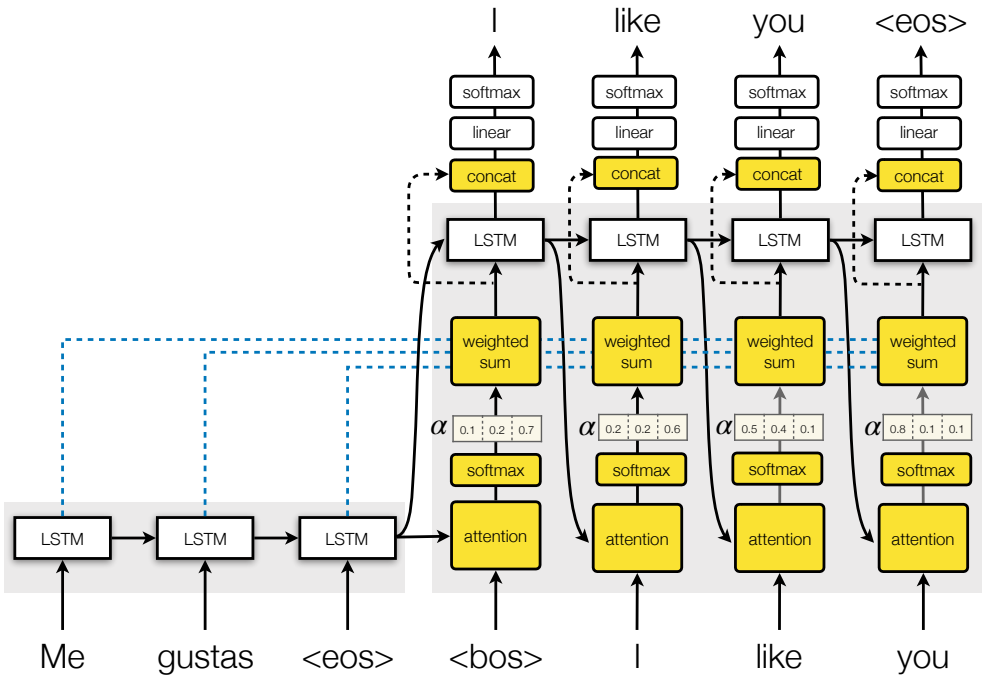

Figure 2.4.: Sequence-to-sequence model with Bahdanau attention.

architecture relies mostly on replicated scaled dot-product attention, referred to as multi-head attention blocks, as shown in Figure 2.5.

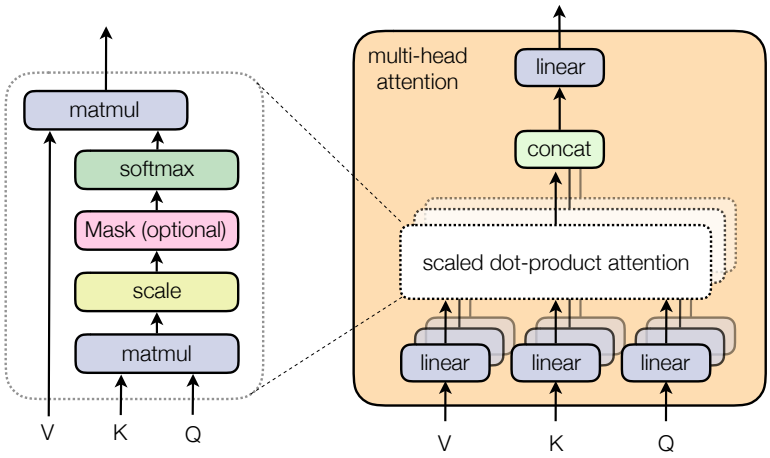

Figure 2.5.: Multi-head attention block.

The Transformer architecture has an encoder-decoder structure where both encoder and decoder consists of stacks of multiple layers of multi-head attention blocks together with layer normalization, residual connections and position-wise feed forward layers. Unlike recurrent units, attention layers do not offer any notion of data sequentiality, so extra positional embeddings are added to the input and output tokens for the attention mechanism to distinguish between the different positions. The model is trained in a completely parallel fashion while at inference time it is autoregressive. In order to ensure the causality of the

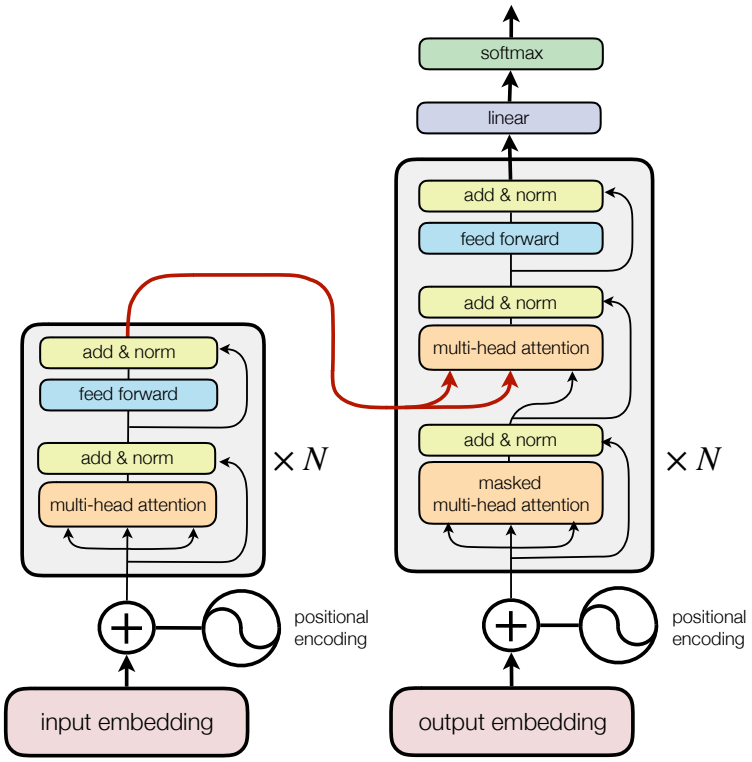

Figure 2.6.: Transformer architecture.

predictions at inference time, during training the decoder self-attention blocks are masked. This architecture is depicted on Figure 2.6.

While there have been several proposals that improve slightly the results of the Transformer model, including the Dynamic Convolution Model (Wu et al., 2019), it is still the most widely used NMT architecture.

Decoding

In an encoder-decoder architecture, at inference time the encoder takes as input a sequence of symbols and generates a sequence of vectors representing such an input sequence, which are then received as input by the decoder. With these vector representation, the decoder generates a probability distribution over the output target language token space, conditioning also on the previous target tokens that have already been predicted. With the output probability distribution, the decoding algorithm must select a token as prediction. The most straightforward way of doing so is to choose the token with highest probability, which is referred to as **greedy decoding**.

However, conditioning greedily only on the previously generated symbols does not necessarily output the sequence with the highest occurrence probability according to the model, as the most likely sequence might not begin with the most likely

symbol. Nevertheless, due to the combinatorial explosion, it is not feasible to evaluate the whole set of possible sequences.

Beam search (Graves, 2012; Sutskever et al., 2014) is a decoding algorithm that relies on the assumption that sequences with high probability have high probability conditionals. It is a form of greedy search where, instead of remembering only the most probable token and condition on it on the next prediction step, keeps the b most probable ones; these are referred to as the *beam*. At the next step, the decoder generates predictions based on the beam from the previous step. It then keeps the top b predictions as the step beam and the process is repeated, only keeping the beam size b elements at each step, and evaluating $b \cdot n$ sequences at each step, where n is the size of the vocabulary. For each end-of-sequence symbol that is selected among the top candidates the beam size is reduced by one and such a translation is added to the final candidate list. When the beam size becomes zero, the search stops. Then, from the final candidate list, the translation with the highest probability, normalized by the number of target words, is chosen.

Evaluation

The evaluation of the translation quality is an open problem that has been faced since the inception of MT. It presents challenges in many different aspects. The first challenge is whether the evaluation is done by humans or automatically. The idealized translation quality evaluation measure is **human evaluation**. However, its high cost makes it undesirable or directly unfeasible at scale. Furthermore, given the subjectivity of translation itself, human evaluation is not very consistent once the quality surpasses certain degree. The optimal approach for humans to evaluate translations is not agreed upon and the most widely known machine translation competition, WMT[1], has switched the human evaluation approach several times (Bojar et al., 2016). Among the most used human evaluation approaches are direct assessment (give a numeric grade to the translation given a reference translation), sentence ranking (rank translations of different MT systems) and evaluation of specific aspects such as fluency and adequacy.

Given its cost, human evaluation is normally dropped in favor of automatic quality measures. Currently, the ubiquitous translation quality measure that dominates both the research and industrial landscapes is the **BLEU score** (BiLingual Evaluation Understudy) (Papineni et al., 2002). It is a based on comparing the candidate translation (hypothesis) with one or multiple reference translations. However, in most cases only one reference translation is considered.

[1] http://www.statmt/wmt20/

BLEU evaluates the *precision* of the translation at the n-gram level. However, some modifications to the computation of the precision are performed in order to avoid some known pathological cases. This way, the computation of the modified n-gram precision of a candidate translation is as follows:

1. For each n-gram and for each candidate translation, count the maximum number of n-gram matches in a single reference translation.
2. For each n-gram and for each candidate translation, clip the total number of matches of a candidate n-gram by the maximal reference match.
3. For each n-gram, add up clipped matches over all candidate translations in corpus.
4. For each n-gram, divide by the total number of unclipped candidate n-gram counts in corpus.

Therefore, the expression that summarizes the precision of the n-grams for a complete test corpus is:

$$p_n = \frac{\sum_{c \in \{candidates\}} \sum_{ngram \in c} count_{clip}(ngram)}{\sum_{c' \in \{candidates\}} \sum_{ngram' \in c'} count(ngram')} \tag{2.3}$$

BLEU combines the individual p_n for different sizes n into a single measure. Given that the modified n-gram precision decays at *exponential rate* with n, the different p_n are not combined with the arithmetic mean, but with the geometric one. An equivalent formulation is to use the arithmetric mean of the logarithms of p_n, using weights w_n to ponderate each term. In the reference implementation, the weights are homogeneous, that is $w_n = 1/N$, for every n, where N is the number of different values of n taken into account.

BLEU also introduces a brevity penalty to discourage candidate translations that have length c that is too short compared to that of the reference translation r:

$$BP = \begin{cases} 1 & c > r \\ e^{1-r/c} & c \leq r \end{cases} \tag{2.4}$$

This way. the final BLEU metric can be expressed as follows:

$$BLEU = BP \cdot e^{\sum_{n=1}^{N} w_n \log p_n} \tag{2.5}$$

The main approach to compute BLEU is the `multi-bleu.perl` script from Moses (Koehn et al., 2007) and `sacrebleu` (Post, 2018), which tries to standarize the evaluation on popular benchmark test datasets.

BLEU has been criticized for its lack of awareness regarding sentence semantics, syntactic structure and morphology, as well as its dependency on a specific tokenization and occasional lack of correlation with human judgement (Babych and Hartley, 2004; Callison-Burch et al., 2006; Tan et al., 2015).

Despite the availability of evaluation methods that try to alleviate these problems, such as METEOR (Denkowski and Lavie, 2014) or ROUGE (Lin, 2004), BLEU is currently the main approach to gauge translation quality.

2.3.3. Neural Language Models

Neural Language Models (LM) compute the probability of a sequence $y_1, \ldots, y_T$ by factorizing it in an autoregressive manner[2]:

$$p(y_1, \ldots, y_T) = \prod_{t=1}^{T} p(y_t | y_1, \ldots, y_{t-1}) \tag{2.6}$$

LMs can therefore be used to estimate the probability of a specific piece of text. Nevertheless, they can also be used as generative models, by using them autoregressively to generate tokens one by one, either from scratch or providing an initial piece of text to be used as context.

Note that LMs can work at any token granularity, with character-level and word-level being the predominant granularities in LMs meant as probabilistic models and subword tokens in LMs meant for text generation (e.g. GPT-2 by Radford et al. (2019)).

In this section we study the different neural architectures used by neural LMs throughout the literature, their use for text generation and finally describe some recent proposals for non-causal LMs.

Neural Architectures

The first attempts of neural LMs were based on feedforward networks (Bengio et al., 2001, 2003) with hyperbolic tangent activations, receiving as input embedded representations of words and using a final projection and a softmax activation to obtain a categorical probability distribution over the token space.

[2] As shown in (2.2) and (2.6), the probability factorization used in LM is the same as the one used in NMT systems, without conditioning the output on a source sentence.

After that, the dominant architectural choices shared the evolution of the decoder part in NMT models: first with recurrent units like LSTMs, shortly followed by convolutional models, with the currently dominant ones being those based on self-attention layers from the Transformer model. Among the recurrence-based LMs, the first applied vanilla RNNs (Mikolov et al., 2010, 2011). Later, the recurrent units were replaced with LSTMs (Zaremba et al., 2014), which were subsequently improved with optimization based on averaged stochastic gradient descent (ASGD) and weight dropping regularization (Merity et al., 2018) (AWD-LSTM). The main difference of these models with respect to recurrence-based decoders in NMT models is their use of "continuous batches" and truncated back-propagation through time (TBPTT): the sentences in the training corpus are concatenated into a single sequence, which is sliced and arranged in mini-batches of N fixed-length sequences so that each sequence at position i in a batch is the continuation of the sentence at the same position in the previously trained batch, and the last hidden state of each sentence is reused as initial state for the next batch, but without propagating the gradients across batches. This enables the network to exploit more context than the used sequence length during training.

Convolutional architectures were also applied to LMs, first directly (Pham et al., 2016) and then with a gated CNN mechanism (Dauphin et al., 2017). These approaches never became dominant and were mostly niche.

The currently dominant architectures are based on self-attention blocks from the Transformer model (Vaswani et al., 2017). They have shown that, with large training data, they are able to achieve state of the art performance in language modeling as well as in multiple downstream tasks using pretrained language models, with the most remarkable example in OpenAI's GPT2 model (Radford, 2018; Radford et al., 2019). The analogous to TBPTT in the Transformer model was proposed in the Transformer XL architecture (Dai et al., 2019), which reuses the inner states of the whole previous batch.

Decoding

LMs can also be used as generative models by using as input their own predictions in an autoregressive manner. However, the actual output of an LM is a probability distribution over the token space. In order to obtain tokens from such a probability distribution there are multiple options. For this, we could apply the same techniques used for NMT, namely greedily take the highest probability token at each timestep, or use beam search. These maximization-based decoding strategies, however, lead to text that is often incoherent or contains word

repetitions. Instead, the state of the art decoding strategies are either to sample among the k most probable tokens (i.e. top k decoding) (Fan et al., 2018) or nucleus sampling (Holtzman et al., 2020), where the tail of the probability distribution is nullified and the tokens are decoded by sampling from the tokens that accumulate most of the probability mass.

Evaluation

Language models can be evaluated either as probabilistic models of text or as text generators.

When evaluated as probabilistic models, the standard measure is perplexity over a test set, which is defined as $2^{-\frac{1}{N}\sum_t^N \log p(x_t|x_{<t})}$, where x_t is each token in the test corpus, $x_{<t}$ are its previous tokens and $p(x_t|x_{<t})$ is the probability computed by the LM. Note that perplexity comparisons are only fair for word-level and character-level LMs, where the token segmentation is unique; on the other hand, subword-level vocabularies cannot be fairly evaluated with perplexity, as there exist multiple valid word segmentations that should be taken into account when computing the probabilities.

When evaluated as unconditional text generators, there is not a standard evaluation procedure. The current trend is to use the language model to generate a piece of text, which is then evaluated in terms of quality and diversity. Quality is normally evaluated by means of the BLEU score over a test set, while diversity is evaluated against the very generated text, evaluating each sentence against every other generated sentence. Given that the probability distribution computed by LMs is normally implemented by a softmax function, it is possible to introduce an extra temperature term τ that regulates the balance between quality and diversity. Therefore, text generation models are normally evaluated for quality and diversity under different values of τ, to understand the performance at every generation regime (Caccia et al., 2020).

Non-Causal LMs

The LM architectures described in previous sections rely on the probability factorization from (2.6), making the token probability predictions causal, that is, each prediction depends on the previous tokens according to the sequential ordering in the sequence.

However, there are other language models that are **non-causal**. The most remarkable example is BERT (Devlin et al., 2019), a masked language model,

where some of the input words are replaced by a special `[MASK]` token at training time, so that the model can learn to guess them. BERT drops the causal mask in transformers' decoder self-attention blocks, thereby predicting masked tokens based on the whole sentence context.

XLNet (Yang et al., 2019), on the other hand, makes use of the Transformer self-attention mask to impose an arbitrary token dependency factorization, potentially completely different from the typical autoregressive one. With this mechanism, it trains the Transformer model with multiple permutations over the tokens' probability factorization, therefore predicting each token with an arbitrary subset of the other tokens in the sentence, aiming at learning more robust representations.

While masked LMs and permutation LMs are only meant to learn representations for transfer learning and not for text generation, they are the foundation of ideas on which non-autoregressive iterative generation approaches rely, and these are directly related to the work presented in this thesis.

2.3.4. Non-sequential Text Generation and Modeling

While traditional language models rely on an autoregressive decomposition of the probability distribution and traditional text generation models are autoregressive, recent lines of research break from such a paradigm and propose novel generation schemes.

Some approaches define variable ordering approaches. Stern et al. (2019) propose the Insertion Transformer, a conditional generative model that iteratively generates pairs of tokens plus the position at which they should be inserted within the sequence, with the ability to generate text from left to right or in a parallel fashion, by decoding according to a balanced binary tree. Emelianenko et al. (2019) simultaneously propose the same approach, going one step further and optimizing the generation order by sampling from the ordering permutations. Chan et al. (2019) propose a similar idea but optimizing a lower bound of the marginalized probability over every possible ordering. A variation of this approach consists in using syntactic information as generation ordering; this is further described in Section 2.5.4.

Other approaches propose a latent variable model where the generation order is treated as latent variable and the training tries to optimize over the generation order space. Gu et al. (2019a) propose to have the generation ordering captured as the relative position through self-attention, optimizing the evidence lower bound (ELBO) to train the model. Gu et al. (2019b) propose Levenshtein

Transformer, a model trained with reinforcement learning to generate token insertion and deletion actions. Welleck et al. (2019) propose a cost minimization imitation learning framework where a policy is learned to generate a binary tree that is used to drive the token generation.

A third non-autoregressive generation paradigm relies on iterative refinement. Lee et al. (2018) propose a latent variable non-autoregressive machine translation model where first the target length is predicted by the model, and then, the decoder is iteratively applied to its own output to refine it. Mask-predict (Ghazvininejad et al., 2019) also predicts the target sentence length and then non-autoregressively predicts the sentence itself, iteratively refining it a fixed number of times, masking out and regenerating the tokens it is least confident about. Lawrence et al. (2019) follow a similar approach and start with a sequence of placeholder tokens (all the same) of a specified length, and they iteratively replace them with normal tokens via masked LM-style inference. As the masking strategy for the training data, the authors propose different stochastic processes to randomly select which placeholders are to be uncovered.

2.4. Linguistic Knowledge and Where to Find It

There are multiple types of linguistic information that can be imbued into neural text generation systems, as well as multiple possible sources of such a kind of knowledge. In this section we describe these options as well as remarkable application examples from the literature.

2.4.1. NLP Formalizations of Linguistic Knowledge

There are several tasks in NLP that directly involve explicit formalizations linguistic knowledge. Among them, the ones most related to the work presented in this thesis are lemmatization, part of speech (POS) tagging, constituency parsing and dependency parsing.

Lemmatization consists in finding the lemma of each word. The lemma is the base form of the word. For instance, in English, the infinitive of a verb is the lemma of a verbal form (e.g. "stop" is the lemma of verbal form "stopped"); and the singular form of a noun is its lemma (e.g. "cat" is the lemma of "cats"). Depending on the degree of morphological complexity of the language, a single lemma may have associated a high number of surface forms.

In POS tagging, each word in the text is associated with a label that denotes the category the word belongs to among a linguistic taxonomy. Some typical examples of such categories in English are noun, verb, adjective, adverb, preposition and conjunction.

Constituency parsing consists in recursively decomposing a sentence into its constituents, casting a tree form that represents the syntactic structure of the sentence.

Dependency parsing consists in identifying the relationships among words in a sentence, obtaining a tree structure with the "head" words and the words modifying them, together with labels that characterize the type of relationship (e.g. "adverbial modifier").

2.4.2. Human-Annotated Corpora

The optimal way to obtain linguistic knowledge is to have human annotators incorporate it to corpora, on which to train the neural systems. However, the effort needed to annotate a corpus is very high and therefore the available human-annotated corpora are small and scarce.

In POS tagging, the Brown Corpus (Francis and Kucera, 1979) is the most frequently used dataset for English.

In constituency parsing, the most relevant corpus for English is the Penn Treebank (PTB; Marcus et al., 1993), specifically a subset made available by Mikolov et al. (2010). This dataset is also frequently used for dependency parsing after applying some transformation rules to obtain dependency parse trees from the constituency trees.

In dependency parsing, the Universal Dependencies initiative (Nivre et al., 2019) compiles dependency parse corpora for multiple languages, all sharing a common set of dependency labels and structures.

2.4.3. Trained Annotation Systems

As human-annotated corpora are very small and scarce, and given the need of neural systems for large training datasets, a practical approach to augment or simply create training data with linguistic knowledge is to train machine learning systems on human-annotated corpora and then use these systems to annotate more data.

As the annotation quality of a trained system is lower than a human, it is expected than the resulting annotated data contains some errors and that it is more noisy.

2.4.4. Rule-based Machine Translation Systems

In Rule-Based Machine Translation (RBMT) systems, the linguistic knowledge itself is formalized in a set of rules and procedures to analyze the source language structures and generate target language ones. This formalization, while not comparable to a human, is designed to generalize well over unseen structures, making RBMT systems more suitable to create synthetic linguistic training data.

One of the sources for linguistic knowledge used in this work is the Rule-based Machine Translation system *Lucy LT* (Alonso and Thurmair, 2003). This tool relies in knowledge distilled and formalized by human linguists in the form of lexicons and rules, and provides a consistent source of linguistic knowledge across several languages, including English, German, Spanish, French, Russian, Italian, Portuguese and Basque. Apart from translations, it provides linguistic analysis byproducts at different levels, which are used here as sources of linguistic information to devise the vocabularies proposed.

The Lucy RBMT system divides the translation process into three sequential stages: analysis, transfer and generation, as illustrated in figure 2.7.

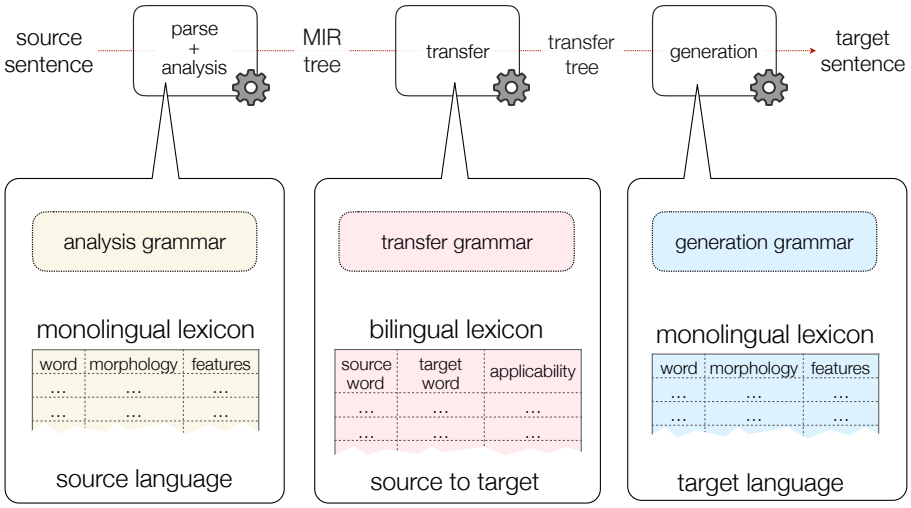

Figure 2.7.: Workflow of rule-based machine translation systems.

The analysis phase receives a sentence in the source language. After being tokenized, the sentence is morphologically analyzed, leveraging a monolingual lexicon to obtain all possible morphological readings of each word in the sentence.

For instance, for the English word "works", the two valid morphological readings are:

"work" (NST) + "s" (N-FLEX)

"work" (VST) + "s" (V-FLEX)

where NST stands for *Noun Stem*, N-FLEX for *Nominal Inflectional Suffix*, VST for *Verb Stem* and V-FLEX for *Verbal Suffix*.

A chart parser together with an analysis grammar converts the sequence of valid morphological readings of the words comprising the sentence and outputs a parse tree. The terminal nodes of the parse tree (i.e. the leave nodes) depend on the monolingual lexicon used during the parse phase. Based on entries in such a lexicon, the parser tries to find inflectional and derivational constructions.

An example of parse tree is shown in figure 2.8.

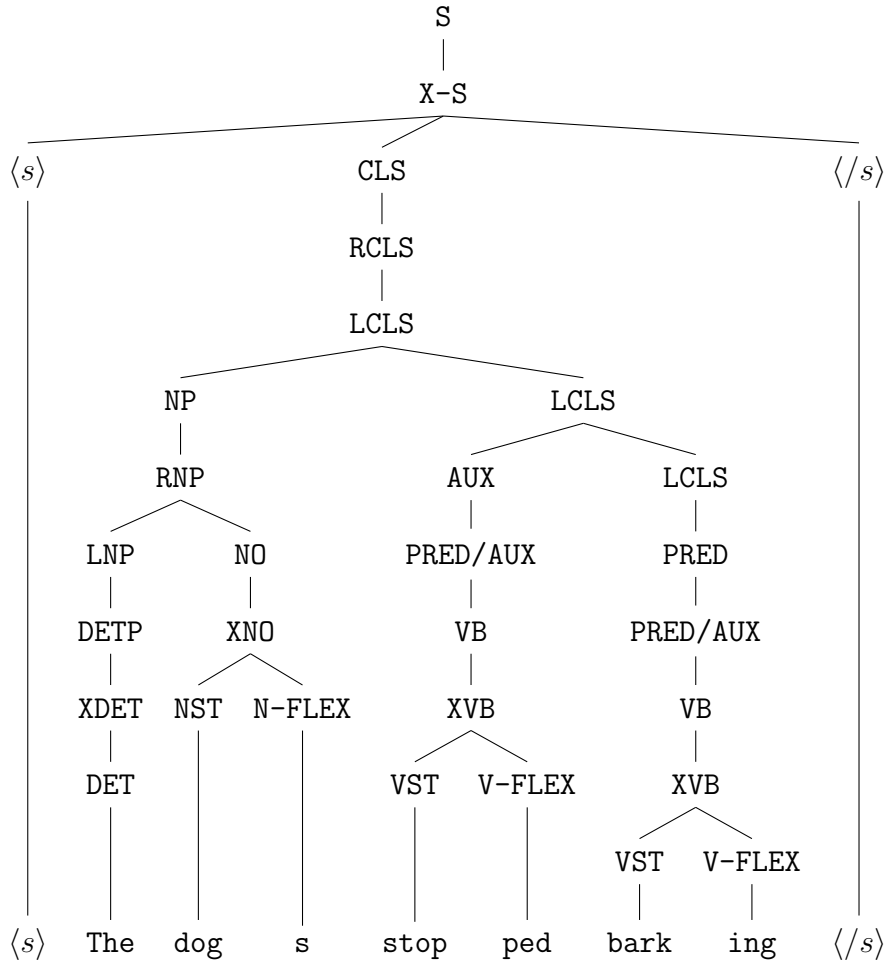

Figure 2.8.: Parse tree for sentence "The dogs stopped barking".

The terminal nodes of the parse tree are the source of the morphological analysis used to create the Morphological Unit Vocabulary described in section 3.1.

The parse three is then applied a second set of rules that annotate, rearrange and mutate the original parse tree nodes, to output an analysis tree, which resembles a projective constituency tree (non projective constructs are rearranged into projective versions). In this tree, words are no longer separated into different nodes representing their morphological parts, but are assembled into a single node with features expressing its morphological traits (e.g. gender, number, verbal tense, person, case).

There is an extra post-processing sub-stage called *mirification* that performs the final retouches, outputting the MIR (Metal Interface Representation[3]) tree. An example of MIR tree is shown in figure 2.9. While there is a noticeable depth reduction in comparison with the parse tree for the same sentence shown in figure 2.8, there are also other non-evident differences: flexions have been merged with their associated lemmas, and the morphological information has been condensed as node features, which are not show in these tree representations.

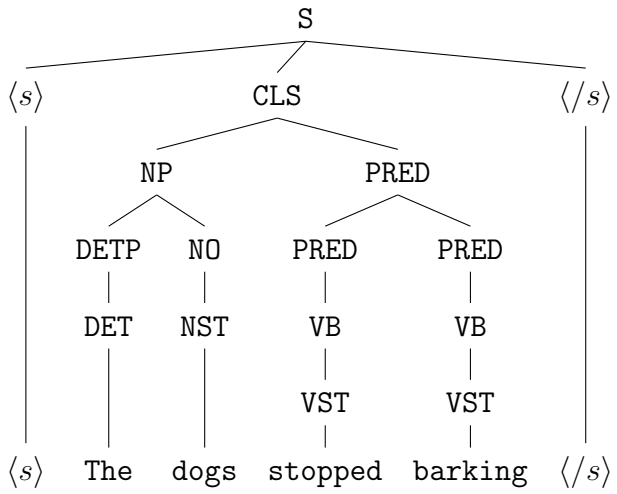

Figure 2.9.: MIR tree for sentence "The dogs stopped barking".

The whole analysis phase is only dependent on the source language and can therefore be reused for language pairs with the same source language. This phase relies in a monolingual lexicon that contains entries for words in the source language, together with meta-information that allows their inflection and morphological derivation. It also relies in an *analysis grammar*, that is, a set of declarative rules that are matched to the input tokens and structures and allow the iterative construction of the parse and analysis trees.

The terminal nodes (i.e. leaves) of the MIR tree are used as the source of the morphosyntactic analysis of the sentence used to create the Lemmatized Vocabulary described in section 3.2. In the MIR tree, terminal nodes represent at

[3]Metal MT is the name of the system developed by the University of Texas and Siemens on which the Lucy RBMT system was initially based (Lamiroy and Gebruers, 1989)

least one word: during the analysis phase, any flexion node is merged with the main word node and such a node gets annotated with morphological features like gender, number, person, tense, case, etc. The presence of these features is language-dependent (e.g. some languages lack case or gender). The morphological features are disambiguated as much as possible taking information from other parts of the sentence (e.g. the person of a verbal form may be disambiguated by the sentence subject). Where not possible, the uncertainty is expressed (e.g. stating all the possible persons the verbal form can be in).

The Lucy analysis takes into account the presence of multi-word expressions (MWE) and handles them as a single element when they are included in the lexicon. This helps capturing the semantics of such constructs during the translation process. This includes not only fixed MWEs (e.g. "in front of"), but also flexible MWEs. For instance, verbal constructions like "take into account" are identified and grouped into a single element.

In the transfer stage, the MIR tree is annotated and mutated into a transfer tree that is suitable as input for the generation phase. There are different types of transfer operations, such as language-pair dependent operations (e.g. mapping of idiomatic expressions), contextual transfer and lexical transfer.

The transfer stage is language-direction dependent. It relies on a bilingual lexicon that contains word and expression translations, together with their context-dependent applicability criteria. It also relies on a *transfer grammar*, that is, a set of imperative rules that implement the needed transformations and annotations.

The generation stage receives as input the transfer tree and generates the final translation, performing any needed reorderings and adaptations. This stage is only dependent on the target language (i.e. it can be reused for any source side language). It relies on a monolingual target language lexicon, together with a *generation grammar*, that is, a set of imperative rules to generate the output sentence.

2.5. Linguistic Knowledge in Neural Models

Since the inception of the first NLP methods, there have been active lines of research trying to imbue linguistic knowledge into them, with aim at improving the system performance. This approach has been applied to symbolic systems, to statistical models and also to neural models.

Linguistic information was first introduced in a neural NLP system by Alexandrescu and Kirchhoff (2006), who proposed an LM where words are represented as a sequence of factors, that is, the word itself plus pieces of linguistic information associated to the word, like its POS tag or the its morphological characterization. Factors of different types are embedded in the same continuous space and the sequence of the previous $n-1$ embedded vectors is fed to the LM, which consists in a multilayer perceptron. The LM then generates the probability of the n-th token over the word space. In order to address the unknown word problem, they compute the average of all words belonging to the same POS tag; this way, if an unknown noun is to be fed to the network, all noun vectors in the embedded space would be averaged to compute the average noun vector.

Another landmark use of linguistic knowledge are recursive neural tensor networks Socher et al. (2013), that allowed to profit from the syntactic structure of the text to drive the composition of representations to address text classification for sentiment analysis.

Since then, multiple proposals have imbued different types of explicit linguistic knowledge into neural NLP systems. In this section we explore those that are most related to the work presented in this thesis.

2.5.1. Linguistically Grounded Vocabularies

Data sparsity is a problem specially affecting morphologically-rich languages, where the amount of different surface forms can be very big to use word-level vocabularies (see Section 2.3.1). To overcome such a problem, some lines of research study how to profit from linguistic information to allow expressing words differently.

Some approaches consist in using automatic annotation tools to obtain linguistic information of the input text, and then using such information to modify the representation of the original words. Following this paradigm, Goldwater and McClosky (2005) study the incorporation of linguistic information to SMT on Czech, which is a highly inflected fusional language and therefore suffers from data sparsity of non frequent surface forms. Their approach consists in lemmatizing low-frequency words and attaching to them extra "pseudo-words" that carried information about the case or the tense, leading to large improvements over word-based baselines. (Tamchyna et al., 2017) propose an NMT system where the decoder, instead of generating words or subword tokens, generates a lemma and a series of morphological tags for it. These are combined in a postprocessing step into the final surface form. Experiments are carried out for

German and Czech, both being morphologically-rich fusional languages, obtaining improvements of up to 1.5 BLEU points over BPE baselines. The linguistic vocabularies we propose in Chapter 3 follow this paradigm.

A different approach is to rely on unsupervised morpheme discovery algorithms, like that of *Morfessor* (Virpioja et al., 2013). Leveraging it, it is possible to obtain morphologically sound word segmentations without an explicit linguistic supervision signal. In that line, Shaik et al. (2011) study different morphologically-grounded subword partition schemes applied to LMs, including morpheme-based, syllable-based and graphone-based, as well as their mix in the same vocabulary with word-based representations for the most frequent words, obtaining improvements of 5% in the word error rate and significat reduction of OOV words. Vania and Lopez (2017) study the effects of subword vocabularies in language models, including BPE and morphologically extracted subwords with *Morfessor*. In their work, the predictions are normal words selected among the most frequent ones, but the input of the model are aggregations of subwords, either by mere addition or by means of biLSTMs. Ataman et al. (2017) and Passban (2017) study different word segmentation strategies and their influence over NMT translation quality, respectively for Turkish and Turkish, German and Russian. They focus on segmenting words into morphologically sound subword units by leveraging Morfessor's unsupervised morpheme discovery. This approach showed to work best on agglutinative languages like Turkish, where independent affixes are added to the word to enrich its meaning, as opposed to fusional languages, where the inflections contain information regarding different semantic aspects, like the case, the gender, etc.

2.5.2. POS Tags, Lemmas and Morphology for NMT

The use of linguistic information was first introduced in NMT in the work by Sennrich and Haddow (2016). In their approach, lemmas, morphological features (case, number and gender for nouns, person, number, tense and aspect for verbs), POS tags and dependency labels are used as linguistic information to enrich the source-side of an NMT system. These pieces of word-level information were attached to each of the subwords belonging to the associated word, for the source-side sentences. Apart from the linguistic information, subwords were tagged with information about whether they are at the beginning, at the middle or at the tail of the word. Hoang et al. (2016) also proposed the factored NMT approach, and studied the effect of different attention variants on it. While the factored NMT approach was tested on the sequence-to-sequence with attention architecture (Bahdanau et al., 2015), Armengol-Estapé et al. (2020) studied its applicability to the Transformer model (Vaswani et al., 2017). All the approaches

mentioned before depend on an external automatic annotation tool to enrich the input text with linguistic information. While some types of linguistic information can be associated to any word type (e.g. POS tags), there are other pieces of morphological information that are specific of certain surface forms. This, depending on how the linguistic information is incorporated in the NMT model, can lead to data sparsity problems. In Chapter 4 we propose sparsely factored NMT, a variant of the approach by Sennrich and Haddow (2016) that is more appropriate for such cases.

While the previous approaches inject linguistic information into the encoder part of the network (i.e. the source side text), it is also possible to do analogously for the target side. In their work, Garcıa-Martınez et al. (2016) proposed to modify the decoder part of a standard word-level sequence-to-sequence model to generate two elements per position of the output sentence: the first element is the lemma of the word, while the second element is the morphosyntactic information of the original word, which is referred to as factors. Each of the two outputs per position casts the probability over the lemma and factor space respectively. A similar approach was proposed by (Song et al., 2018) for the Russian language; they modify the decoder of a normal sequence-to-sequence with attention model to generate first the stem of the current word, and then its suffix based on the internal states and output of the decoder units, and then using a composite loss with a separate terms for stems and for suffixes.

The generation of proper surface forms of morphologically rich languages has been studied in the literature, especially in transduction from morphologically simpler languages (e.g. English-to-German translation). With that purpose, Conforti et al. (2018) proposed to predict the morphological information of a morphologically rich language from merely the lemmas and word capitalization scheme.

2.5.3. Dependency LMs

The use of dependency parse trees to drive a language model was first proposed by Chelba et al. (1997), with a similar structure to an n-gram LM, but where the context of a word is its preceding bigram plus a list of preceding words whose parent does not precede it. Their model was not generative, but was only meant to compute the perplexity of an input sentence. They suffered from the same problem as our Iterative Expansion LMs (see Chapter 6), namely that they need both a sentence and a candidate dependency parse tree to compute the perplexity. In order to be able to approximate the unconditional perplexity (which would need to marginalize over all possible dependency trees for the given sentence),

the authors use the product of the probability of the sentence given the parse tree and the probability of the parse tree given the sentence.

Shen et al. (2008) make use of the dependency tree in a probabilistic LM, as part of an SMT system. They computing the probability of each word conditioned on its parent and the sibling words between both. Their approach is also an approximation to the perplexity based on heuristics. Given that the purpose of their LM was just to compare different sentences generated by the SMT system, the value of the perplexity was not the focus, but rather the relative value for different candidate translations.

Mirowski and Vlachos (2015) propose a dependency LM based on RNNs, where the dependency tree is decomposed into a collection of unrolls, that is, paths from the root to one of the leaves, and where the probability of a word can be predicted from these unrolls. In order to enable the computation of the sentence-level perplexity, they assume that each word in a sentence is conditionally independent of the words outside of its ancestor sequence in the dependency parse tree. Buys and Blunsom (2018) propose a shift-reduce transition-based LSTM (Hochreiter and Schmidhuber, 1997) dependency LM that can be used for language modeling and generation by means of dynamic programming.

In the early experiments of Iterative Expansion LMs (see Section 6.7), we discarded their use to compute perplexities because the approximations we experimented with drew values that were not comparable with sequential LMs, which do not need marginalizing to obtain perplexities.

2.5.4. Syntax-driven Generation and Modeling

While most text generation model are autoregressive, there are several recent proposals for non-autoregressive generation (see Section 2.3.4). Normally, those approaches either define explicitly the generation ordering or handle the generation order as latent variables and try to optimize over the insertion ordering space. There are other approaches that try to assimilate the mentioned latent variables with syntactic information, either dependency parse trees or constituency trees. Following this paradigm, recurrent neural network grammars (RNNG; Dyer et al., 2016) are recursive models that operate with a stack of symbols that can be populated with terminals or nonterminals or "reduced" to generate a syntactic constituent, obtaining as a result a sentence and its associated constituency parse tree. In the same line, syntactically supervised transformers (Akoury et al., 2019) make use of a simplified form of the constituency parse tree as latent variables, modeling it autoregressively in a supervised way to later use it as input for a fully non-autoregressive transformer that generates the output

sentence. Our proposal for Iterative Expansion LMs from Chapter 6 follows this same paradigm.

Other models try to model syntactically sound tree structures while avoid a syntactic supervision signal. Shen et al. (2018) propose parsing-reading-predict networks, where skip-connections are used to integrate constituent dependency relations with RNNs. Their model does not need syntactic supervision but can learn the underlying dependency structures by leveraging a syntactic distance together with structured attention. Ordered neurons (Shen et al., 2019) are a modified version of LSTMs where the latent sentence tree structure is used to control the dependencies between recurrent units by means of special "master" input and forget gates.

3. Linguistic Knowledge-based Vocabularies for Neural Machine Translation

In neural networks for text generation, text is normally represented as discrete tokens, either with one token per word (word-level vocabulary) or splitting individual words into subwords and representing each subword with a differen token (subword-level vocabulary). Further background on discrete text representation in neural networks can be found in section 2.3.1.

In this chapter we propose two different strategies that rely on linguistic information to provide morphologically sound vocabulary definitions for their use in neural networks applied to NLP tasks. As illustrated in Figure 3.1, we propose to use a linguistic engine, which is described in detail in section 2.4.4. Such an engine was used in both the vocabulary extraction phase, were the vocabulary is defined based on a training corpus, and in the token encoding phase, where the vocabulary is leveraged to encode the text as token identifiers. Note that the vocabulary extraction phase takes place before training the network and the token encoding phase takes place both at training time (to encode the training texts) and at inference time.

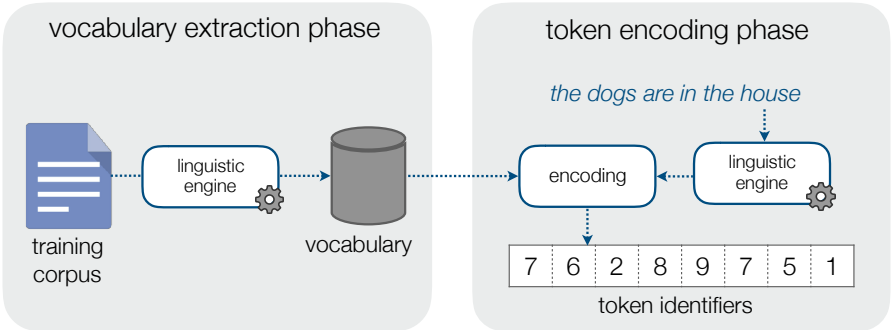

Figure 3.1.: Vocabulary extraction and token encoding phases.

In the following sections we describe both the vocabulary extraction phase and the token encoding phase for each of the two proposed approaches. Related work in the application on linguistic knowledge for vocabulary creation can be found in section 2.5.1.

3.1. Morphological Unit Vocabulary

The goal of the Morphological Unit Vocabulary is to serve as a linguistically-grounded subword vocabulary, aiming at addressing the out-of-vocabulary problem of word-level vocabularies while allowing a morphological interpretation of the segmentation. This vocabulary definition strategy relies on the morphological analysis of a sentence, which comprises a sequence of morphological units that may be lexical morphemes, multi-morpheme stems, separate inflectional morphemes or even fixed/semiflexible multi-word expressions, e.g. "in front of".

During vocabulary extraction, all sentences in the training data are analyzed and their morphological units are used to elaborate the vocabulary, as shown in Figure 3.2. The specific information from the node that is incorporated as a token comprises the string associated with the node (being it a lexical morpheme, a word or a multi-word expression), together with its category, which is loosely analogous to the Part-of-Speech (POS) tag (e.g. noun stem (`NST`), verb stem(`VST`), noun flexion (`N-FLEX`)).

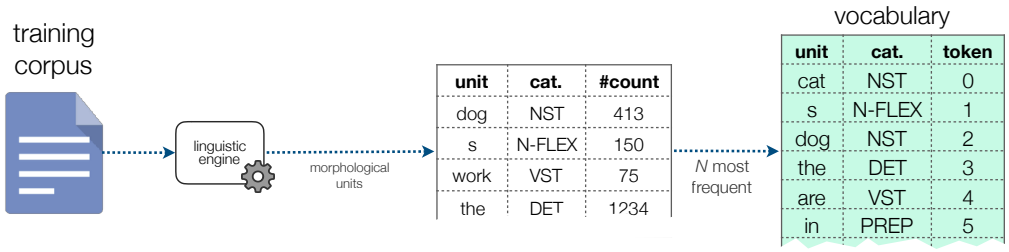

Figure 3.2.: Morphological subword vocabulary extraction.

In order to encode a text into a sequence of tokens, the text is analyzed by means of a linguistic engine and the resulting morphological units are used as queries to find the associated token indexes from the vocabulary table.

Given the high amount of possible tokens and the practical size limitations of a vocabulary meant to be used with neural networks, only the N most frequent tokens from the training data are selected to be part of the vocabulary.

If the analysis is driven by a lexicon, like in our case, this constrained vocabulary implies a mismatch with the unconstrained vocabulary used by the linguistic engine: when encoding the tokens of a text, the parse tree may contain terminal nodes that we cannot encode because they are not part of the vocabulary, either because they were not present in the training data or because their frequency of appearance was not enough to grant an entry in the final size-limited vocabulary. In order to eliminate such a vocabulary mismatch, once the Morphological Subword Vocabulary is extracted, the lexicon used by the linguistic engine (which

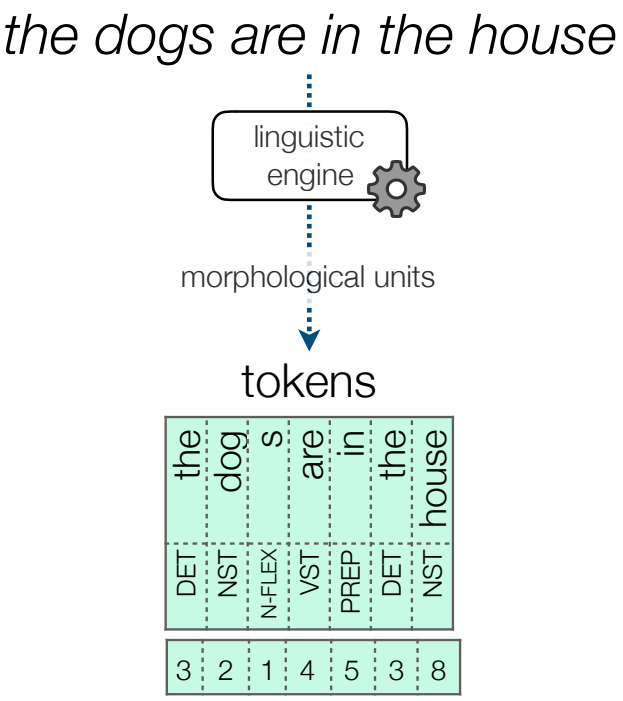

Figure 3.3.: Token ID encoding process with the morphological unit vocabulary.

drives the extraction of the morphological units) is pruned to remove any entry that is not part of the extracted vocabulary. These results in the removal of low-frequency words that, if encountered during the token encoding of a text, will be encoded as unknown words.

Words that are not part of the training data are marked in the analysis as unknown words. In order to cope with this OOV word situation, we can follow the approach by Luong and Manning (2016) and reserve some of the tokens in the vocabulary for character-based tokens. This way, any character found in the training data has its own token in the reserve character-based token range. As with subword vocabularies, this character-based subvocabulary makes `<UNK>` tokens not necessary for Mophologic Unit Vocabularies.

The resulting layout of the tokens table is outlined in Figure 3.4, with an initial range for special tokens like the end of sequence token or the padding token, an optional small range for character-level tokens, and finally the largest range for the morphological unit tokens.

Some examples of the resulting Mophological Unit tokenization are:

- *The dogs are in the house*: (the, `DET`), (dog, `NST`), (s, `N-FLEX`), (are, `VST`), (in, `PREP`), (the, `DET`), (house, `NST`), ⟨ /s ⟩

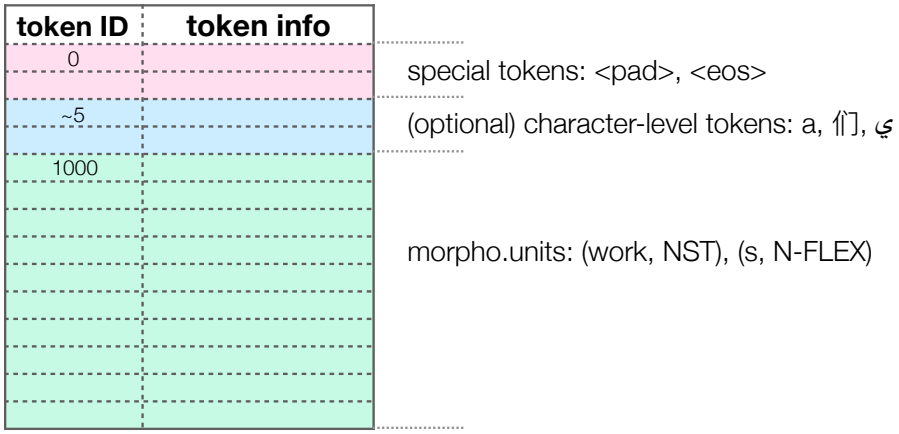

Figure 3.4.: Overall distribution of the morphological units vocabulary table.

- *My mom said I mustn't tell lies*: (my, `DET`), (mom, `NST`), (sai, `VST`), (d, `V-FLEX`), (I, `PRN`), (must, `VST`), (n't, `ADV`), (tell, `VST`), (lie, `NST`), (s, `N-FLEX`) ⟨/s⟩

3.2. Lemmatized Vocabulary

The goal of the Lemmatized Vocabulary is to decouple meaning from morphological information in each word. For this, each word generates two tokens: one for the lemma and one for the relevant morphological traits of the word (e.g. gender, number, tense, case).

The source of linguistic information in this case is the morphosyntactic analysis of the sentence, which provides information for each word about its POS tag and its morphological features, such as gender, number, person, tense, case, etc. The presence of these features is language-dependent (e.g. some languages lack case or gender). Note that the morphological features do not contain information about the semantics of the word, but only about the morphological traits that, when added to the lemma, conform the specific surface form of the word.

During the vocabulary extraction phase, all sentences in the training data are analyzed and the resulting lemmas and morphological features are used to elaborate the vocabulary, as shown in Figure 3.5. For each word, the lemma is added to a lemma frequency counter, and the morphological features are added to an analogous morphological feature-set frequency counter.

In order to encode a text into a sequence of tokens, the text is analyzed by means of the linguistic engine. For each word, we obtain the lemma and the set of its morphological features (e.g. verb in present tense first person singular). For each lemma and for each morphological feature set we then query the vocabulary table

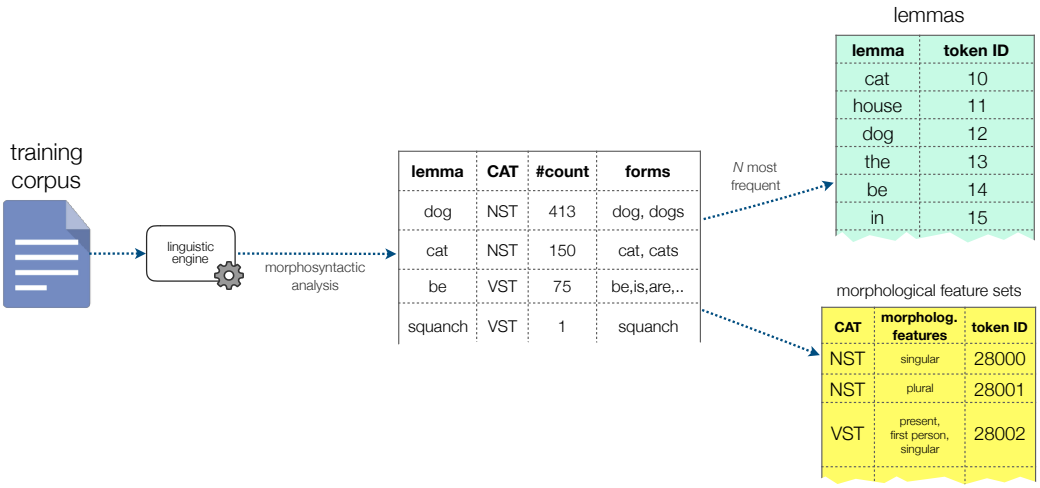

Figure 3.5.: Lemmatized Vocabulary extraction.

for the appropriate token ID. This is illustrated in Figure 3.6, where the reuse of morphological feature set token IDs is highlighted in bold font.

As in the Morphological Unit Vocabulary (see Section 3.1), the mismatch between the Lemmatized Vocabulary and the lexicon used for the morphosyntactic analysis is solved by pruning the latter to only contain elements from the former. The same way, unknown words are encoded by allocating a range of the token indexes for character-based tokens and using such character-based subvocabulary to encode any string that is marked as unknown. The distribution of the different elements present in a Lemmatized Vocabulary is illustrated in Figure 3.7.

In order to cope with out-of-vocabulary words, we reserve a range of tokens for character-level tokens so that any word or numeral can be encoded whether it was seen or not in the training data. The layout of the Lemmatized Vocabulary table is outlined in Figure 3.7, where we can see an initial range for special tokens, an optional range for character-based tokens, the largest range for the lemma tokens and the final range for every possible morphological feature set found in the training data. Note that another possibility to address the OOV words is to add the special token `<UNK>` to represent them and have a post-processing step to handle such a token; a frequent approach is to use the attention vector of sequence-to-sequence models to replace any `<UNK>` token at the output with the word from the input sentence with the highest attention value.

The nature of the linguistic engine we use gives us a morphosyntactic analysis with some deviations from the original sentence: first, the words in the sentence are rearranged to turn its structure into a projective parse, if it was not projective already. This way, the English sentence “Who do you want me to talk

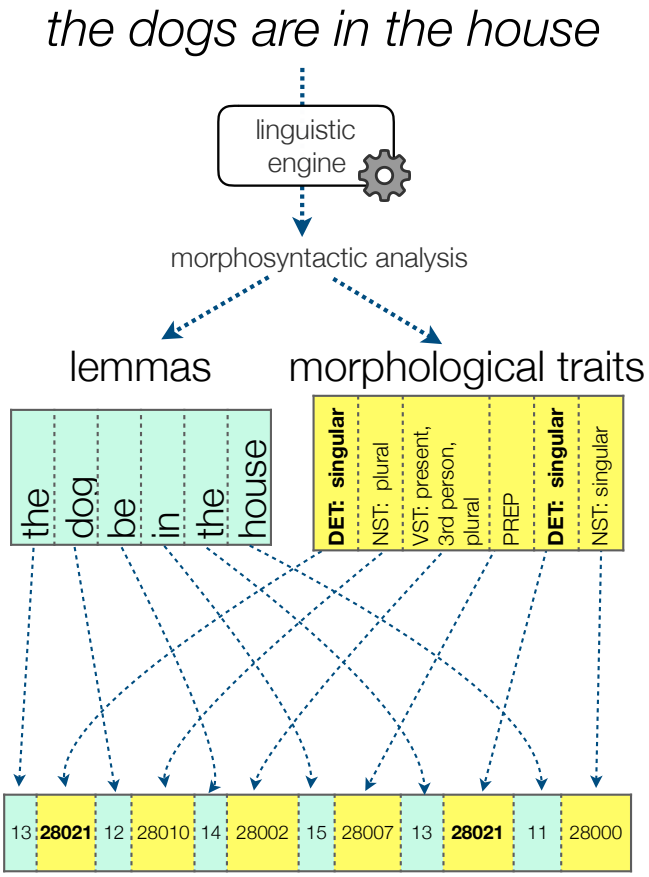

Figure 3.6.: Token encoding phase with the lemmatized vocabulary.

to?" is rearranged as "You do want me to talk to who?". A similar rearrangement occurs for other cases like separable phrasal verbs, which are rearranged so that the preposition sits next to the verb, and both form together a single multiword; this way "You let me down" would be rearranged into "You let down me", and "let down" would be a single entity, with a single lemma and a single morphological feature set. This word rearrangements and aggregations favor a semantical interpretation of the sentence when used to represent the input to a neural system.

Given that the morphological information tokens always follow the lemma tokens, and that there are words in natural languages that do only admit one surface form, the lemmatized vocabulary can waste tokens that add no further information. In order to avoid such a situation, we only include the morphological information tokens if they are actually needed, that is, if the lemma they are associated to admits more than one surface form and hence can be subject to morphological variations.

Some examples of the resulting Lemmatized tokenization are:

- The dogs are in the house

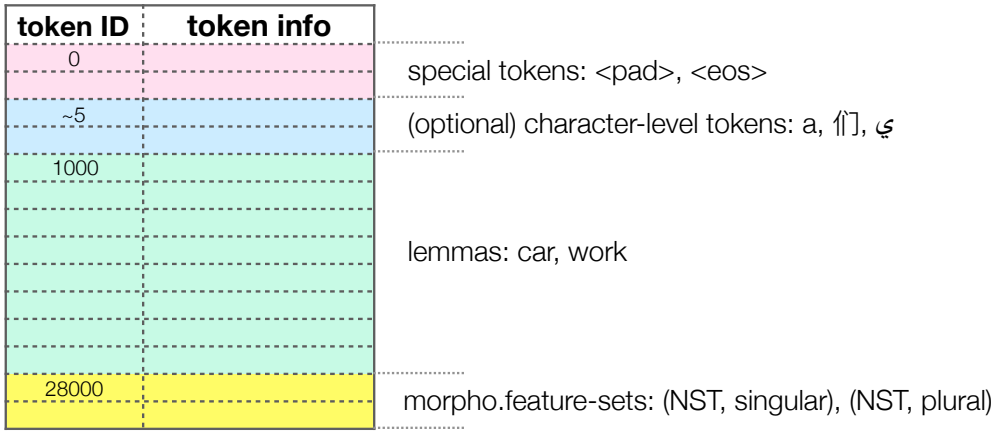

Figure 3.7.: Overall distribution of the lemmatized vocabulary table.

lemma: the, *morpho*:(`DET:(NU (PL SG))`),
lemma: dog, *morpho*:(`NST:(NU (PL) PS (3))`),
lemma: be, *morpho*:(`VST:(MD (IND) NU (PL) PF (FIN) PS (3)...)`),
lemma: in, *morpho*:(`PREP:()`),
lemma: the, *morpho*:(`DET:(NU (PL SG))`),
lemma: house, *morpho*: (`NST:(NU (SG) PS (3))`),
⟨/s⟩

- My mom said I mustn't tell lies:

lemma: my, *morpho*: (`DET:(NU (PL SG))`,
lemma: mom, *morpho*: (`NST:(NU (SG) PS (3))`,
lemma: say, m*orpho*: (`VST:(MD (IND) NU (SG)...)`,
lemma: I, *morpho*: (`PRN:(CA (S) NU (SG) PS (1))`),
lemma: must, *morpho*:(`VST:(MD (IND) NU (SG)...)`,
lemma: not,
lemma: tell, *morpho*:(`VST:(MD (IND) NU (SG PL)...)`,
lemma: lie, *morpho*:(`NST:(NU (PL) PS (3))`,
⟨/s⟩

3.3. Experiments

In order to evaluate the vocabulary definition strategies proposed in Sections 3.1 and 3.2, we test them using machine translation as downstream task.

Neural Machine translation models compute the translation of a source sequence of tokens $x_1, \ldots, x_T$ by predicting token by token of the translation sequence $y_1, \ldots, y_{T'}$, which has a potentially different length T':

$$p\left(y_1, \ldots, y_{T'} | x_1, \ldots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | x, y_1, \ldots, y_{t-1}\right) \tag{3.1}$$

The currently dominant NMT architecture is the Transformer model (Vaswani et al., 2017), which surpasses in translation quality the original sequence to sequence models (Sutskever et al., 2011; Cho et al., 2014) and their variants with attention (Bahdanau et al., 2015; Luong et al., 2015). In our NMT experiments, we make use of the original implementation of the Transformer architecture by their authors, who released it as part of the `tensor2tensor` library. We use a standard configuration (`transformer_base`), which can be found in the hyperparameter configuration shown in Table 3.1, with independent embeddings in the encoder and decoder inputs in order to freely allocate the source embedding table for the linguistic vocabulary and, in the target side, with the final projection tied with the embedding matrix. We also used parameter averaging after convergence.

attention layers	6
attention heads per layer	8
hidden size (embedding)	512
batch size (in tokens)	4096 ($\times$ 4 GPU)
training steps	20 epochs
vocabulary type	word pieces
vocabulary size	32K
optimization algorithm	Adam
learning rate	warmup + decay

Table 3.1.: Hyperparameters of the Transformer model for the NMT experiments.

We performed experiments on English-German, French-English and Basque-Spanish datasets. The purpose of choosing those languages is to test the proposed vocabulary definition strategies both in morphologically rich languages (i.e. Basque, German) and in morphologically simpler ones (i.e. English).

German nouns are inflected for number (singular and plural), gender (masculine, feminine and neuter) and case (nominative, accusative, genitive and dative). French nouns are inflected for number (singular and plural) and gender (masculine and feminine). English nouns are only inflected for number (singular and plural) and case (nominative and genitive). Spanish nouns are inflected for number (singular and plural) and gender (masculine and feminine). Basque nouns

are inflected (or rather they take suffixes for) number (singular, plural and "mugagabe") and case (nominative, ergative, genitive, local genitive, dative, allative, inessive, partitive, etc.).

As far as verbs are concerned, German verbs have different inflections for 1st, 2nd and 3rd person singular and 1st/ 3rd persons and 2nd person plural in the present. French verbs are inflected for number and person, and gender in perfective compound tenses. English finite present tense verbal forms are only inflected in the 3rd person singular. Spanish verbs are inflected for person (1st, 2nd and 3rd), number (singular and plural), tense (present, past, future), aspect (perfective, punctual and progressive) and mood (indicative, subjunctive, conditional and imperative). Basque verbs take different forms for person (1st, 2nd and 3rd, not only for the subject but also for the direct and indirect objects), number (singular and plural), tense (present, past and future), aspect (progressive and perfect) and mood (indicative, subjunctive, conditional, potential and imperative).

Also, German presents compounds, that is, concatenation of words with no separation in between:

`Übersetzungsqualität` → `Übersetzung` (translation) + `s` + `Qualität` (quality)

`Speicherverwaltung` → `Speicher` (memory) + `Verwaltung` (management)

For the English-German experiments, we make use of the WMT14 English-German news translation data[1]. The characteristics of the used training dataset are summarized in Table 3.2.

Corpus	Sents.	Words	Vocab.	Max.length	Avg.length
German	4520620	96159821	3181111	2937	21.3
English		103664418	1909854	4225	22.9

Table 3.2.: Statistics of the German-English training data.

For the French-English experiments, we make use of a combination of the News Commentary corpus and the Europarl corpus. The characteristics of the resulting training corpus are shown in Table 3.3

Corpus	Sents.	Words	Vocab.	Max.length	Avg.length
French	2085044	64894699	145953	245	31.1
English		58984908	117311	237	28.3

Table 3.3.: Statistics of the French-English training data.

[1] http://www.statmt.org/wmt14/translation-task.html

For the Basque-Spanish experiments, we use the EiTB news corpus (Etchegoyhen et al., 2016). Its characteristics are shown in Table 3.4.

Corpus	Sents.	Words	Vocab.	Max.length	Avg.length
Basque	552752	10102635	345351	318	18.3
Spanish		15643597	225038	317	28.3

Table 3.4.: Statistics of the Basque-Spanish training data.

In order to evaluate the translation quality, we use BLEU (Papineni et al., 2002), which consists of an aggregation of n-gram matches together with a penalty for sentences shorter than the reference translations. The BLEU scores shown were computed by means of the `sacrebleu` tool (Post, 2018) with the lower case setting. Given the known problems BLEU presents (Callison-Burch et al., 2006), we also include the METEOR (Banerjee and Lavie, 2005) scores, except for Basque, which is not supported by METEOR.

Vocabulary	de-en		en-de		
	BLEU	METEOR	BLEU	METEOR	
word pieces	31.81	0.3537	26.35	0.4800	(baseline)
(Sennrich and Haddow, 2016)	30.20	0.3386	25.90	0.4653	(baseline)
lemmatized	31.14*	0.3521	25.49*	0.4697	
morpho.units	31.33*	0.3505	25.89*	0.4764	

Table 3.5.: German-English and English-German translation quality (case-insensitive BLEU score) with different source vocabulary strategies (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline).

Vocabulary	fr-en		en-fr		
	BLEU	METEOR	BLEU	METEOR	
word pieces	32.01	0.3554	34.36	0.5707	(baseline)
(Sennrich and Haddow, 2016)	27.60	0.3288	31.90	0.5430	(baseline)
lemmatized	29.66*	0.3404	33.68	0.5677	
morpho.units	31.30*	0.3516	34.82	0.5758	

Table 3.6.: French-English and English-French translation quality (case-insensitive BLEU score) with different source vocabulary strategies (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline).

Vocabulary	eu-es		es-eu		
	BLEU	METEOR	BLEU	METEOR	
word pieces	28.89	0.5072	24.48	-	(baseline)
(Sennrich and Haddow, 2016)	24.16*	0.4654	21.45*	-	(baseline)
lemmatized	27.32*	0.4945	22.39*	-	
morpho.units	28.52	0.5045	23.83*	-	

Table 3.7.: Basque-Spanish and Spanish-Basque translation quality (case-insensitive BLEU score) with different source vocabulary strategies (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline).

In Tables 3.5, 3.6 and 3.7 we can see the BLEU scores obtained by using different source vocabulary definition strategies, for German↔English, English↔French and Basque↔Spanish respectively. As baselines, we used a word piece vocabulary (Wu et al., 2016) and the linguistic factored approach by Sennrich and Haddow (2016). The word piece vocabulary was used for the original implementation of the Transformer model (Vaswani et al., 2017), with the same hyperparameter configuration from Table 3.1, with shared embeddings in the encoder and decoder inputs, and also in the final projection[2]. The factored

approach by Sennrich and Haddow (2016) is the standard way for incorporating linguistic information; we used the same extra linguistic features as the authors, namely the lemma, POS tag and syntactic dependency label; as a subword vocabulary is used, each feature is copied to all subwords in the same word, and the position of the subword within the word (beginning, end, middle) is also added as feature; all feature embeddings are concatenated together with the token embedding to form the subword representation. In order to make this baseline comparable to the word piece baseline and to our own work, we added the linguistic features to the Transformer model instead of the original LSTM-based sequence-to-sequence with attention model from (Sennrich and Haddow, 2016), keeping all the hyperparameters from the word piece baseline, while using the same linguistic feature-related hyperparameters from (Sennrich and Haddow, 2016), namely the feature embedding dimensionalities. We used the implementation of the factored NMT Transformer from `OpenNMT-py` (Klein et al., 2017) with custom improvements in order to support specifying vocabulary sizes and embedding dimensions for the linguistic features. We used the base hyperparameter configuration (see Table 3.1), with separate embeddings for source and target sides, in order to freely allocate the source embedding space among the factors. In the target side the output projection was tied with the embeddings.

[2] https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py

For the linguistic annotations we used Stanford's `corenlp` (Manning et al., 2014) for English and French, `ParZu` (Sennrich et al., 2009, 2013) for German, like in the original work by Sennrich and Haddow (2016), LucyLT (Alonso and Thurmair, 2003) for Basque and Spacy (Honnibal and Montani, 2017) for Spanish. Note that the Morphological Units and Lemmatized vocabularies include the character-level subvocabulary described in Section 3.1 to handle OOV words.

In all cases, the target language vocabulary strategy are word pieces in order to ensure a proper comparison.

As part of the experiments carried out, we also evaluate the influence of the proposed morphologically-based vocabularies on the translation quality for out of domain texts. For this, we use the WMT17 biomedical test sets, namely the English-German HimL test set[3] the French-German EDP test sets[4], and a sample of 1000 sentences of the Open Data Euskadi IWSLT18 corpus (Jan et al., 2018), which contains documents from the Public Administration.

Given that these benchmarks are not included in `sacrebleu`, we used Moses' `multi-bleu.pl` script, together with the standard tokenizer. The out-of-domain results are summarized in Tables 3.8, 3.9 and 3.10.

In order to assess the statistical significance of the differences between our proposed approaches and the word pieces baselines for the in-domain and out-of-domain test, we made use of the bootstrap resampling approach (Koehn, 2004; Riezler and Maxwell, 2005)[5], taking 95% as significance level ($p < 0.05$ in the hypothesis test comparing with the word piece baseline). Statistical significance is reflected in the result tables with a * mark next to the BLEU score.

Vocabulary	**de-en**		**en-de**		
	BLEU	METEOR	BLEU	METEOR	
word pieces	40.77	0.4059	36.75	0.5547	(baseline)
(Sennrich and Haddow, 2016)	37.64	0.3723	33.86	0.5160	(baseline)
lemmatized	41.35*	0.4059	36.04	0.5496	
morpho.units	41.57*	0.4076	36.67	0.5549	

Table 3.8.: German-English and English-German translation quality in out-of-domain text (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline)

[3] http://www.himl.eu/test-sets

[4] https://www.statmt.org/wmt17/biomedical-translation-task.html

[5] Moses script `bootstrap-hypothesis-difference-significance.pl` was used to compute the significance tests.

Vocabulary	fr-en		en-fr		
	BLEU	METEOR	BLEU	METEOR	
word pieces	16.85	0.2122	19.58	0.3763	(baseline)
(Sennrich and Haddow, 2016)	14.89	0.1993	18.02	0.3607	(baseline)
lemmatized	15.74*	0.2086	18.34*	0.3681	
morpho.units	16.25	0.2146	19.36	0.3749	

Table 3.9.: French-English and English-French translation quality in out-of-domain text (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline)

Vocabulary	eu-es		es-eu		
	BLEU	METEOR	BLEU	METEOR	
word pieces	16.94	0.4439	5.78	-	(baseline)
(Sennrich and Haddow, 2016)	13.80*	0.3715	7.01*	-	(baseline)
lemmatized	19.85	0.4348	8.75	-	
morpho.units	20.66	0.4423	9.06	-	

Table 3.10.: Basque-Spanish and Spanish-Basque translation quality in out-of-domain text (* $p < 0.05$ in the hypothesis test comparing with the word piece baseline)

The obtained English↔German results suggest that, while for the morphologically poor language (English) the translation quality is the same as the strong subwords baseline, the quality for the morphologically rich language (German) is improved in a statistically significant way. On the other hand, for English↔French results are weaker in the case of the lemmatized vocabulary, while the morphological units vocabulary presents comparable performance to the word pieces baseline. For Basque and Spanish, we see a very large improvement of both lemmatized and morphological unit vocabulary, with up to 3.5 BLEU points more than the word pieces baseline for Basque→Spanish and 3.2 BLEU points for Spanish→Basque. We conclude that for the morphologically poor language, the use of linguistic vocabularies actually harms the translation quality for in-domain data (Tables 3.5, 3.6, 3.7), while for a morphologically rich language there is statistical evidence that the quality is higher than the strong subword baseline for out-of-domain data for German and Basque and comparable for French (Tables 3.8, 3.9, 3.10). This way, for the morphologically rich language with in-domain test data and for the morphologically poor language with out of domain data there is no statistical evidence to distinguish the quality of our proposed approaches from the strong subword baseline.

Baseline	(...) and were treated in intensive care stations
Morpho.units	(...) and were treated in intensive **care units**
Reference	(...) and were receiving care in intensive **care units**
Baseline	(...) pest printing was regularly monitored
Morpho.units	(...) the **skull pressure** was regularly monitored
Reference	(...) had regular monitoring of **pressure in the skull**
Baseline	Our objective was to investigate whether the number of people who died changed by the appointment of antithrombin.
Morpho.units	Our objective was to investigate whether the number of people who died changed by **administering** antithrombin.
Reference	Our goal was to investigate whether the number of people who died changed by **giving antithrombin**.
Baseline	it is not known whether the peripheral Iridium inhibits the development or progression of a pigment plum in practice.
Morpho.units	it is not known whether peripheral **irridotomy** inhibits the development or progress of pigment **glaucoma**.
Reference	it is unknown whether peripheral **iridotomy** reduces the development or progression of pigmentary **glaucoma**.
Baseline	(...) the use of Neuamine inhibitors
Morpho.units	(...) the use of **neuraminidase** inhibitors
Reference	(...) the use of **neuraminidase** inhibitors

Table 3.11.: German-to-English out-of-domain examples.

Table 3.11 shows some examples comparing the German-to-English outputs from out-of-domain text of the baseline and the Morphological Unit Vocabulary. The examples show that our linguistically-driven morphological segmentation has a clear impact on choosing more appropriate lexical units. Improvements come either from infrequent or specific words (e.g. glaucoma, irridotomy) or from generic words that are adequate for the particular context (e.g. units, administering).

3.4. Discussion

The proposed linguistic knowledge-based vocabulary definition strategies offer a way to profit from morphosyntactic information for downstream tasks like MT. The two main differences with other approaches like factored NMT (Sennrich and Haddow, 2016) derive from the use of a semantics-aware linguistic engine and from its non-aggregative management of linguistic information.

About the linguistic engine used, given that its ultimate goal is to perform rule-base translation, it needs to analyze the semantics of the input sentence, and uses it to disambiguate when multiple possible interpretations of a word are possible. When the disambiguation is not possible (e.g. when the subject of a sentence is not present and the verb conjugation admits more than one interpretation), the uncertainty is reflected in the analysis and our proposed vocabularies use such an information to compose the encoded representation. Another peculiarity of the used linguistic engine is that its analyses are driven by a lexicon. This makes it possible to adjust it to match the neural vocabulary in order to avoid mismatches between word and multi-word representations in both sides.

The non-aggregative encoding strategy makes it possible for the systems addressing the downstream tasks to directly use linguistic information, but also makes the resulting sequences longer. In order to further characterize the impact in sequence length, we computed the distribution of the ratio of the sequence lengths of both the Morphological Unit Vocabulary and the Lemmatized vocabulary with respect to a normal space and punctuation-based tokenization. The vocabularies are extracted from the training data, while the distribution is computed over a sample of 1000 sentences of the same dataset. We compute such a distribution for a configuration of our vocabularies where the OOV words are encoded as an `<UNK>` token and also where they are handled by a character-level subvocabulary, in order to understand the influence of this type of words over the final sequence length. The distribution of the same ratio for a word pieces vocabulary is also computed as reference. Figure 3.8 shows the distributions for the Morphological Unit Vocabulary, while Figure 3.9 shows it for the Lemmatized Vocabulary.

As we can see in Figures 3.8 (Morphological Units) and 3.9 (Lemmatized), the sequence length with the proposed morphologically-grounded vocabularies with respect to the number of words in the sentence is higher than with word pieces (Wu et al., 2016), especially when the character-level subvocabulary is used to cope with the OOV words.

As shown in the figures, the differences in length depend on the morphological characteristics of the specific language. For English, with a simpler morphology, the ratio of sequence length with the proposed morphology-based vocabularies with respect to word pieces is higher than with German, French or Basque, which have richer morphology and hence needs also more word pieces for a single sentence.

This difference in length may affect the quality depending on the model's ability to handle long-range dependencies. For instance, multi-head attention mechanisms are known to be able to properly handle such type of dependencies, while RNNs present problems in that regard (Hochreiter, 1991; Bengio et al., 1994).

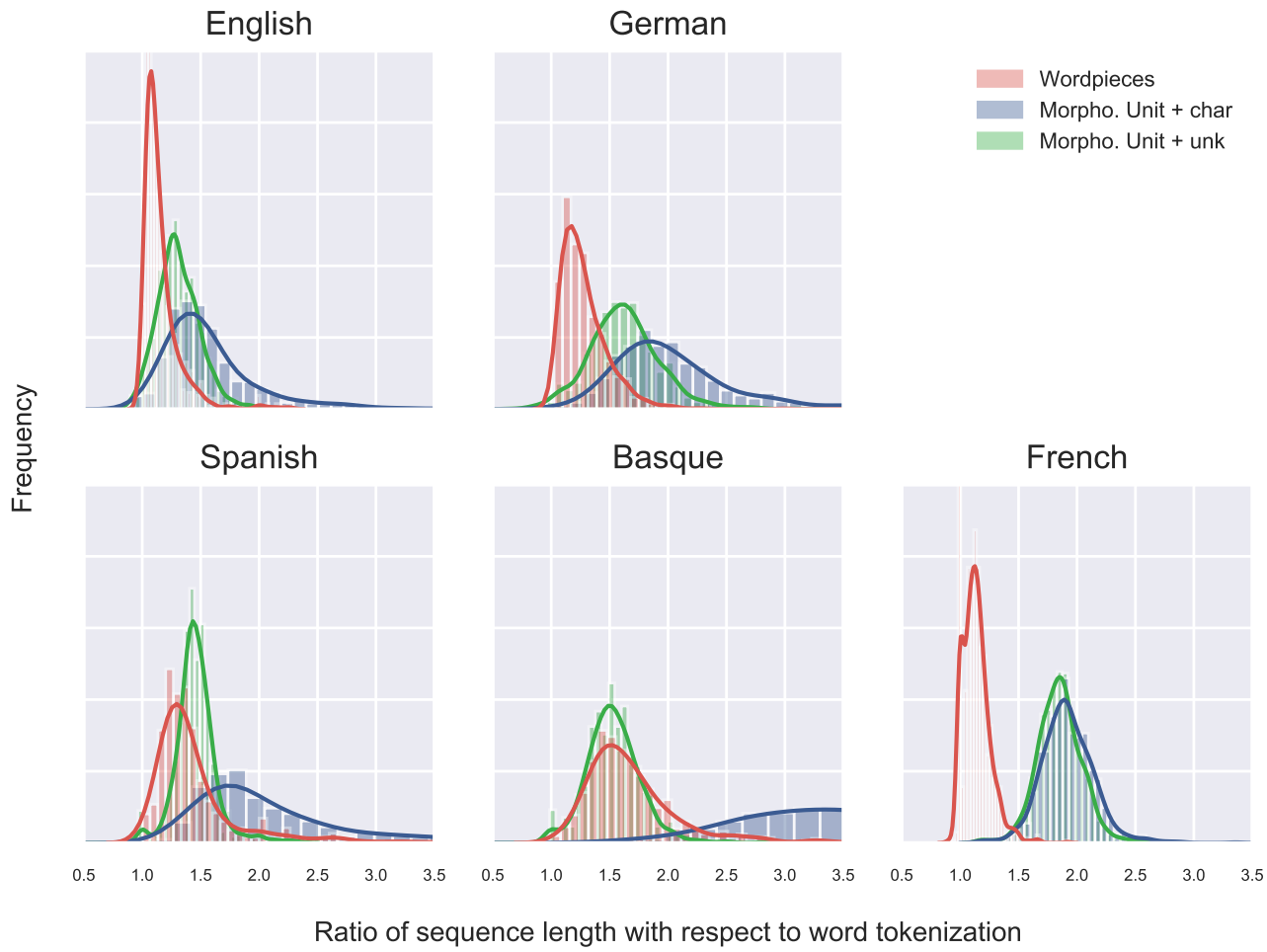

Figure 3.8.: Distribution of the ratio of sequence length with the Morphological Unit Vocabulary and a standard word-based tokenization.

The non-aggregative encoding strategy also allows using neural architectures without any modification, unlike the factored approaches like those by Sennrich and Haddow (2016) and Garcıa-Martınez et al. (2016), which need to account for the different representation spaces for lemmas and factors and keep separate embedding tables, which multiply the number of hyperparameters to tune, namely the vocabulary size and embedding dimensionality for each of the linguistic features. In this sense, the results obtained by factored approaches using the same hyperparameter configuration as Sennrich and Haddow (2016) offer inferior translation quality compared to the word piece vocabulary; this can be attributed to the non optimality of the hyperparameters for our specific datasets and the usage of the Transformer architecture instead of the original LSTM sequence-to-sequence with attention model from (Sennrich and Haddow, 2016).

Therefore, compared to word piece approaches and to the linguistic approach by Sennrich and Haddow (2016), the mophological vocabularies approach is suitable for scenarios where the source language is a morphologically rich language like

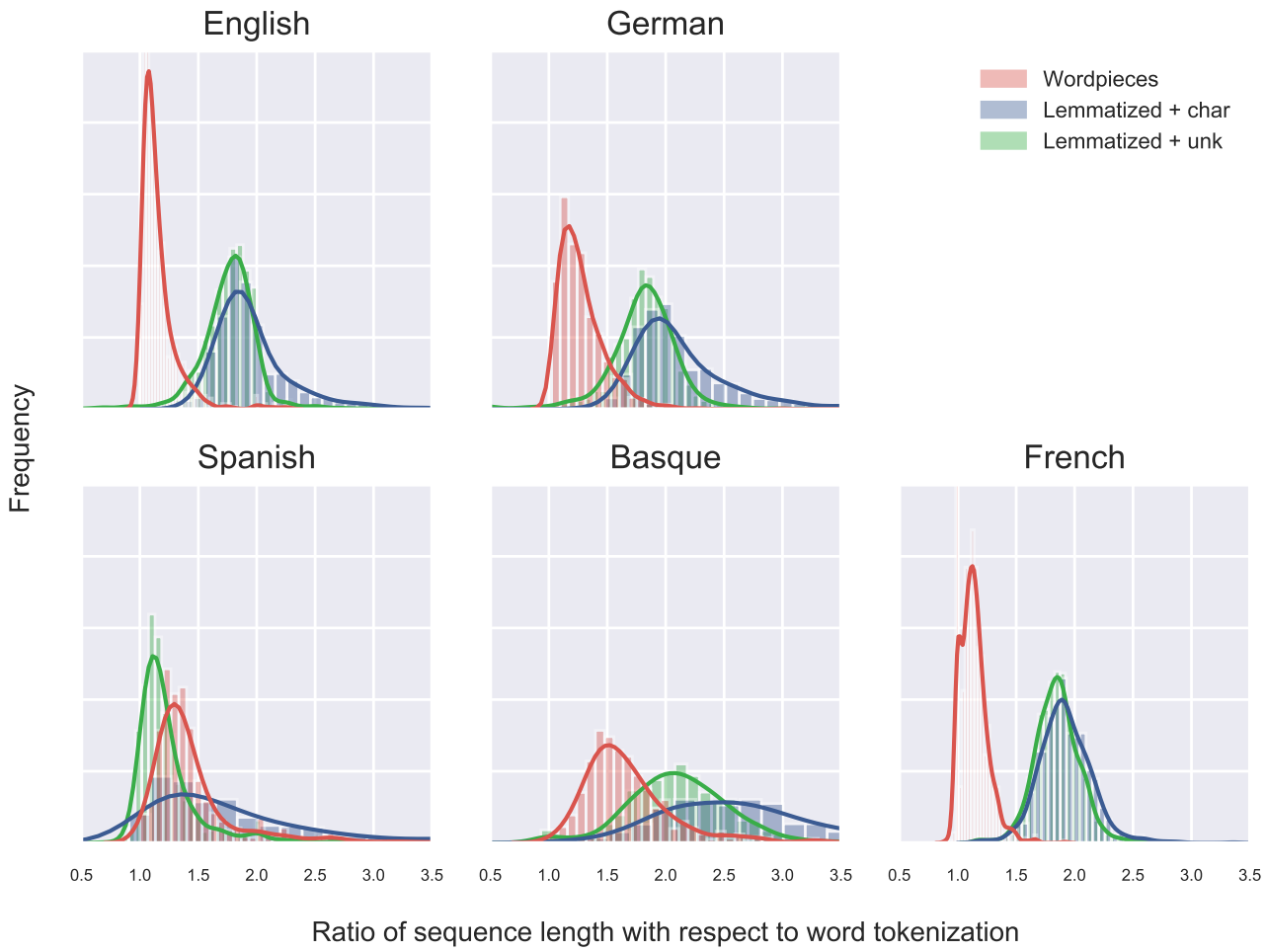

Figure 3.9.: Distribution of the ratio of sequence length with the Lemmatized Vocabulary and a standard word-based tokenization.

German, where the chosen neural architecture can handle long-range dependencies, like the Transformer model (in order to cope with the longer sequences), and where the available training data does not match the domain of the text the model is going to be fed as input at inference time.

3.5. Conclusion

Our experiments show that the proposed morphology-based vocabulary definition strategies provide improvements or maintain comparable quality in the translation of out-of-domain texts for languages that present a rich morphology like German and Basque. We also observe that no significant loss is suffered in translation quality for morphologically poor languages like English in that type of texts.

Qualitatively, whenever we inject linguistic information in our neural systems, we are progressing in the interpretability of such systems. In this chapter we proposed to do a linguistically-driven segmentation of our vocabulary, which enables morphologically-aware interpretation of the performance in downstream tasks. This is a line of research to be pursued in the future, especially in relation to the use of linguistic vocabularies for text generation, for instance, using the proposed vocabularies for the target side in NMT tasks.

4. Sparse factored Neural Machine Translation

Domain shift is one of the main challenges yet to overcome by neural machine translation (NMT) systems (Koehn and Knowles, 2017). This problem happens when using an MT system to translate data that is different from the data used to train it, mainly regarding its domain (e.g. the MT system was trained on news data but is then used to translate biomedical data). The problem consists in a drop in the translation quality with respect to translations of in-domain text.

Injecting linguistic information has been used in the past to improve the translation quality of NMT systems (see Section 2.5 and Chapter 3). The improvements obtained for in-domain data are normally small, while those obtained for out-of-domain text are usually larger (Casas et al., 2020c; García-Martínez et al., 2020). The most frequent and straightforward approach to inject linguistic information into NMT systems is to use annotation systems to obtain word lemmas and part-of-speech (POS) tags. These pieces of information are then attached as "factors" to each subword in the original word Sennrich and Haddow (2016). In this scheme, however, it is assumed that each word has a value for each of the possible factors. We refer to these as "dense" linguistic annotation schemes.

Nevertheless, not all linguistic annotations are dense. Some examples of morphologically rich language features that are not dense include noun cases and verb conjugations, where only some type of words can be tagged with such kind of information. These "sparse" linguistic annotation schemes cannot be easily accommodated in factored NMT architectures, as the space of possible values of the morphological features factor is large (each word can have a combination of such feature values) and a specific combination may seldom appear in the training data, despite the fact that each of its individual feature values may appear frequently. This leads to a situation where many of the embedded vectors of the morphological features factor are updated infrequently during training. This fact is illustrated in Figure 4.1, where we show the frequency count of the morphological feature combinations versus the frequency count of each individual morphological feature, for the training split of one datasets used in our experiments. In that figure, we can appreciate that the number of different combinations is an

order of magnitude larger than the individual features (580 combinations vs. 24 individual feature values), and that the frequency count is also multiple times lower.

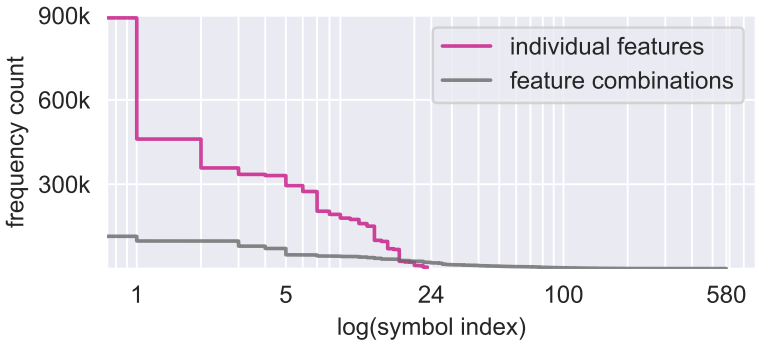

Figure 4.1.: Presence of the morphological information attributes in the training data words. The grey histogram reflects the frequency count of the different combinations of the morphological features factors in the IWSLT14 de-en German data, while the purple histogram reflects the frequency count of each morphological feature value with the same textual data. The morphological features were extracted with `ParZu`. The X axis is expressed in logarithmic scale.

In this chapter, we propose an approach to inject sparse linguistic annotations into NMT systems. We refer to it as sparse factored NMT. Related work in the application on linguistic knowledge for vocabulary creation can be found in section 2.5.1.

4.1. Sparse factored NMT

In our proprosed approach, instead of taking raw text as input to translate, like normal NMT systems do, we receive the text annotated by a linguistic annotation system for the source-side. For training, in the target side we take raw text. This aspect is the same as in the work by Sennrich and Haddow (2016). However, in their work, for sparse linguistic annotation schemes, like the morphological features they use, each annotation is a collection of attributes that may or may not be present for each word type.

Instead of taking each combination as a different factor value, we propose to label each word based on the morphological feature space instead of the morphological feature combination space. For this, we keep an embedding table where each entry is a value of a morphological feature. For instance, in the German sentence "Wir brauchen Daten, keine Hilfe", taken from the IWSLT14 validation data, in

factored NMT pronoun "Wir" would be labeled with the morphological feature combination `1|Pl|_|Nom` (first person, plural, nominative case), while in sparse factored NMT the same word would be labeled with three tags: `1`, `Pl` and `Nom`.

Also, regarding the use of word tokenization or subword tokenization (e.g. Byte-Pair Encoding (BPE; Sennrich et al., 2016c)), we propose the following. Apart from the morphological feature vocabulary described before, we also maintain a lemma-vocabulary; when encoding text for sparse factored NMT, for each word we check if it is a lemmatizable word (i.e. not a number, punctuation, etc) and if its lemma is present in the lemma vocabulary. If it is, we encode the word as the embedded vector of the lemma. This vector is added to the embedded vectors of every morphological feature the word had to compute the word's vector representation.

If the word could not be lemmatized or if the lemma is not present in the lemma vocabulary, we tokenize the word using BPE, for which we keep also an embedding table. In this case, there are no morphological features incorporated, just the subword token representation.

Therefore, our tokens can be either lemma plus morphological features or subwords. Once the text is encoded as a sequence of embedded vectors, it is passed as input to a standard Transformer model (Vaswani et al., 2017). Note that our proposal only affects the embedding layer of the encoder of an NMT architecture. Therefore, it can be applied to both sequence-to-sequence with attention or the Transformer.

We propose a further extension on top of the base variant described before: we take a new hyperparameter, the "linguistic dropout" (LD), which represents the probability of using a subword tokenization for a word instead of the (lemma + morphological features) representation. During data preparation, both the subword representation and the lemmatized representation (if available) are prepared and, during training, a sample of the Bernoulli distribution with the LD probability determines which representation is used for each word at batch creation time. The purpose of LD is to make the model learn to handle the situation where there is no linguistic information available (e.g. for out-of-vocabulary words). Using LD, the subword token embeddings are more frequently updated during training, leading to more robust systems, especially on out-of-domain data.

4.2. Experimental Setup

In our experiments, we make use of morphologically-rich languages, namely German and Basque, with low-resource scenarios, testing with both in-domain and

out-of-domain data.

German is a West Germanic Language with fusional morphology. Its nouns are inflected in terms of number (singular and plural), gender (masculine, feminine and neuter) and case (nominative, accusative, genitive and dative). Verbs inflect for person (1st, 2nd and 3rd), number, mood (indicative, imperative, subjunctive, infinitive), voice, tense (present, preterite, perfect, pluperfect, future, future perfect), grammatical aspect, and completion status. For the German experiments, we use the IWSLT14 German→English dataset (Cettolo et al., 2014) as training data. Its statistics are shown in Table 4.1. For the in-domain translation quality evaluation, we used the mentioned dataset test split, while for out-of-domain translation evaluation we used the WMT17 biomedical test sets, namely the English-German HimL test set[1]. The preprocessing used was the one recommended by fairseq for the IWSLT14 de-en data[2], namely corpus cleaning, tokenization and lowercasing with Moses scripts (Koehn et al., 2007), and the BPE subword vocabulary had 10k merge operations.

Corpus	Sents.	Words	Vocab.	Max.len.	Avg.len.
German	160k	3.1M	113k	172	19.4
English		3.3M	53k	175	20.4

Table 4.1.: IWSLT14 German-English training data stats.

Basque is a language isolate (not related to other languages), with agglutinative morphology. Its nouns take suffixes to express number (singular, plural, "mugagabe") and case (nominative, ergative, genitive, local genitive, dative, allative, inessive, partitive, etc). Verbs' surface forms differ based on the person of the subject, direct object and indirect object (1st, 2nd and 3rd), number (singular and plural), tense (present, past, future), aspect (progressive and perfect) and mood (indicative, subjunctive, conditional, potential and imperative). For the Basque experiments, we use the EiTB news corpus (Etchegoyhen et al., 2016). Its statistics are shown in Table 4.2. We split the original data[3] into training, validation and test subsets. The test split was used for in-domain translation quality evaluation, while a sample of 1000 sentences of the Open Data Euskadi IWSLT18 corpus (Jan et al., 2018) containing documents from the Public Administration, was used for the out-of-domain translation evaluation. The preprocessing for the data consisted in truecasing and tokenization (this preprocessing was already applied in the original data), and the BPE subword vocabulary had 15k merge operations.

[1] http://www.himl.eu/test-sets

[2] https://github.com/pytorch/fairseq/tree/master/examples/translation

[3] https://aholab.ehu.eus/metashare/repository/browse/basque-spanish-eitb-corpus-of-aligned-comparable-sentences/5f5bd836b6f111e6b004f01faff11afa8b95c93ec1214a338167e5074ee90d09/

Corpus	Sents.	Words	Vocab.	Max.len.	Avg.len.
Basque	550k	10.1M	345k	318	18.3
Spanish		15.6M	225k	317	28.3

Table 4.2.: EiTB Basque-Spanish training data statistics.

The linguistic information used for the experiments was obtained with Lucy LT (Alonso and Thurmair, 2003), a rule-based machine translation (RBMT) system of transfer type. We took the analysis of the source sentences generated in the intermediate stages of the translation, which annotates each word with a bag of linguistic language-specific features, covering all the morphological and grammatical traits of the word.

We train sparse factored NMT systems with and without linguistic dropout. For LD, we used $p = 0.25$, that is, there is a 75% probability of using the (lemma + morphological features) representation, if available, and 25% probability of using the word's subword tokens instead. The neural architecture used for out experiments was the Transformer model, using the base hyperparameter configuration of fairseq and a batch of 4096 tokens.

We included two baseline systems as reference. First, a vanilla Transformer model with BPE vocabulary and shared encoder and decoder embeddings (and, in the target side, the final projection before the softmax being tied to the input embedding) without any linguistic information. Second, a factored NMT system (Sennrich and Haddow, 2016). Both baselines use the same hyperparameters: for IWSLT14 de-en, we use the hyperparameters recommended by fairseq for that dataset while, for EiTB eu-es, we use the hyperparameters of the base transformer, but with a smaller total batch size of 4096. For the German linguistic information we used the `ParZu` annotation tool (Sennrich et al., 2009, 2013) (which was the tool used by Sennrich and Haddow (2016)), while for Basque we use the analysis by Lucy LT. For the factored NMT, we used OpenNMT (Klein et al., 2017) (which supports token features) with its implementation of the Transformer. For the vanilla Transformer we used fairseq (Ott et al., 2019). For the sparse factored NMT system, we created a custom implementation on top of fairseq's Transformer.

Given the different input and output vocabularies, the embedding tables in encoder and decoder of the models with morphological information, both ours and the baselines, were not shared. In the target side, though, the input embedding is tied with the final projection before the softmax.

In our experiments, we studied the translation quality in terms of BLEU scores (Papineni et al., 2002), obtained with Moses' `multi-bleu.perl` script after tokenizing with the Moses tokenizer. Given that our datasets had been true-

cased/lowercased, we compute the lower-case variant of the BLEU score (flag `-lc` of `multi-bleu.perl`).

The hyperparameter tuning was done manually, trying a less than 8 configurations, focusing on the dropout, linguistic dropout and number of attention heads. All experiments were performed on a server with 4 nvidia 1080Ti GPUs.

4.3. Results

Table 4.3 shows the BLEU scores obtained by our sparse factored NMT, with and without linguistic dropout, as well as the baseline systems, for the German (DE) → English (EN) and Basque (EU) → Spanish (ES) translation directions, both with in-domain and out-of-domain tests.

	eu→es		**de→en**	
MODEL	IN DOMAIN	OUT OF DOMAIN	IN DOMAIN	OUT OF DOMAIN
Without linguistic info	29.8	19.9	34.8	3.2
Factored	24.2	13.8	32.0	8.4
Sparse factored	28.6	19.7	32.6	8.0
Sparse factored + LD	29.4	20.7	34.3	9.2

Table 4.3.: Translation quality (case-insensitive BLEU scores) of the proposed model (Sparse factored NMT, with and without linguistic dropout) and baseline models: BPE without linguistic information and Factored NMT.

We can see that the factored NMT system in general performs worse than the Transformer baseline without linguistic information. This can be associated with the sparsity problem described in Section 4.1 and illustrated in Figure 4.1, which is especially relevant for an agglutinative language like Basque, where the difference for in-domain data is 5.6 BLEU points.

We can also appreciate that with sparse factored NMT without LD, we also suffer a loss in translation quality with respect to the vanilla Transformer. However, using sparse factored NMT, we have comparable translation quality with respect to the vanilla Transformer for in-domain data, but for out-of-domain data we improve 0.8 BLEU points for Basque and 6 BLEU points for German.

From these results, we understand that, without LD, the subword token embeddings are under-trained. This problem is mitigated by the introduction of LD. The results also suggest that the improvements can be larger in very low resource scenarios, like the German experiments, with 160k sentences in the training data,

a much smaller size than Basque, with 550k. For in-domain data, our approach suffers a small loss, 0.4-0.5 BLEU points, which is normally considered comparable.

4.4. Conclusion

We proposed sparse factored NMT, which is an approach to inject linguistic information in the source-side of NMT architectures, especially appropriate for annotation schemes where the morphological tags are not applicable to all word types, leading to sparseness of the training signal in classical approaches like factored NMT. We also proposed linguistic dropout, a complement to sparse factored NMT that improves the training signal for the subword embeddings.

Our experiments showed that this approach maintains the baseline translation quality, only with a minor loss, and improves drastically the translation quality of out-of-domain text when the system has been trained in a low-resource setting.

Future work may include detailed analyses of the specific influence of some hyperparameters, like the sharing or not of the encoder and decoder embeddings, over the final translation quality, as well as a qualitative analysis of the differences in the outputs of the different studied models, including linguistic constructions that are better handled by one or the other.

5. Combining Subword Representations into Word-level Representations

Currently dominant NMT architectures receive as input sequences of discrete tokens taken from fixed-size source and target token vocabularies defined a priori. Before being fed to the network, the input text is tokenized and the positions of those tokens within the vocabulary table are the actual network inputs.

As commented in detail in Section 2.3.1, the granularity of the tokens in those vocabularies can range from character-level, to subword-level, to word-level.

Character-level token granularity, while allowing maximum representation ability with minimal vocabulary size for alphabet-based scripts, also delegates word formation modeling to the network and makes token sequences to be much longer than with word-based tokens.

Using word-level tokens leads to very large vocabulary sizes, especially for morphologically rich languages, where the number of surface forms per lemma is high. Large token vocabularies are impractical for the current neural architectures and hardware so it is frequent to constrain the vocabulary size to a few tens of thousand tokens, which is hardly enough to fit the number of symbols in a complete word-based vocabulary; compositional word structures like numbers pose further problems with such a granularity level, as well as proper nouns. When word-based vocabularies are used, the vocabulary is built with the most frequent surface forms in the training data, which normally leads to degradation of translation quality.

Subword-level token granularity offers a compromise between representational power and vocabulary size, especially statistically extracted subword vocabulary strategies like Byte Pair Encoding (BPE) (Sennrich et al., 2016c).

Models with word-level token vocabularies can incorporate word-level information as extra input to the model by combining it one-to-one with the token representations. Some examples of word-level information are Part of Speech (POS) tags, syntactic dependency relationships or lemmas. In order to make use of word-level information in models with subword-level token vocabularies, a usual approach is to assign the word information to all its subwords (Sennrich

and Haddow, 2016). This approach, despite improving the translation quality, introduces an information assignment mismatch, that is, the high-level linguistic information belonging to the whole word is combined with low-level subword token information, sometimes with subwords being a single letter inside a long word.

In this chapter, we propose to modify the Transformer architecture (Vaswani et al., 2017) to combine the learned subword representations into word representations in the encoder block. This allows to naturally incorporate any extra word-level information directly at the level of word-level representations.

The contents of this chapter are structured as follows: the proposed approach is described in section 5.1, while the experimental setup is presented in section 5.2 and the results are described and discussed in section 5.3. Finally, the conclusions are drawn in section 5.4. Also, the relevant related work is described in sections 2.3.1 and 2.5.2.

5.1. Subword to Word Transformer

In the standard Transformer architecture from Vaswani et al. (2017), the encoder applies a series of self-attention layers to the input token embeddings. The output of the encoder is then used at every layer of the decoder as key and value of the multi-head attention. In these operations, the token representations in the sequences in the source batch are masked according to the original sequence lengths in tokens.

We propose to divide the encoder into two blocks of self-attention layers. The first block receives the embedded subword-level token representations and processes them through $N_{sw}^{(e)}$ layers of self-attention like those from the nominal Transformer. The subword-level representations obtained as result of the first block are then combined into word level representations (different combination strategies were studied, being described later in this section). A second block of $N_{w}^{(e)}$ self-attention layers processes these word-level representations. The output of the second encoder block is then fed to the first $N_{w}^{(d)}$ layers of the decoder, while the following $N_{sw}^{(d)}$ decoder layers are fed with the output of the first block of the encoder. The appropriate padding masks are used in the decoder depending on whether the encoder output used is subword or word-level. This architecture is shown in Figure 5.1.

In our first tests we directly used the encoder word representations as keys and values to every decoder layer (instead of using the encoder subword representations in the last layers of the decoder). This, however, led to poor results. We

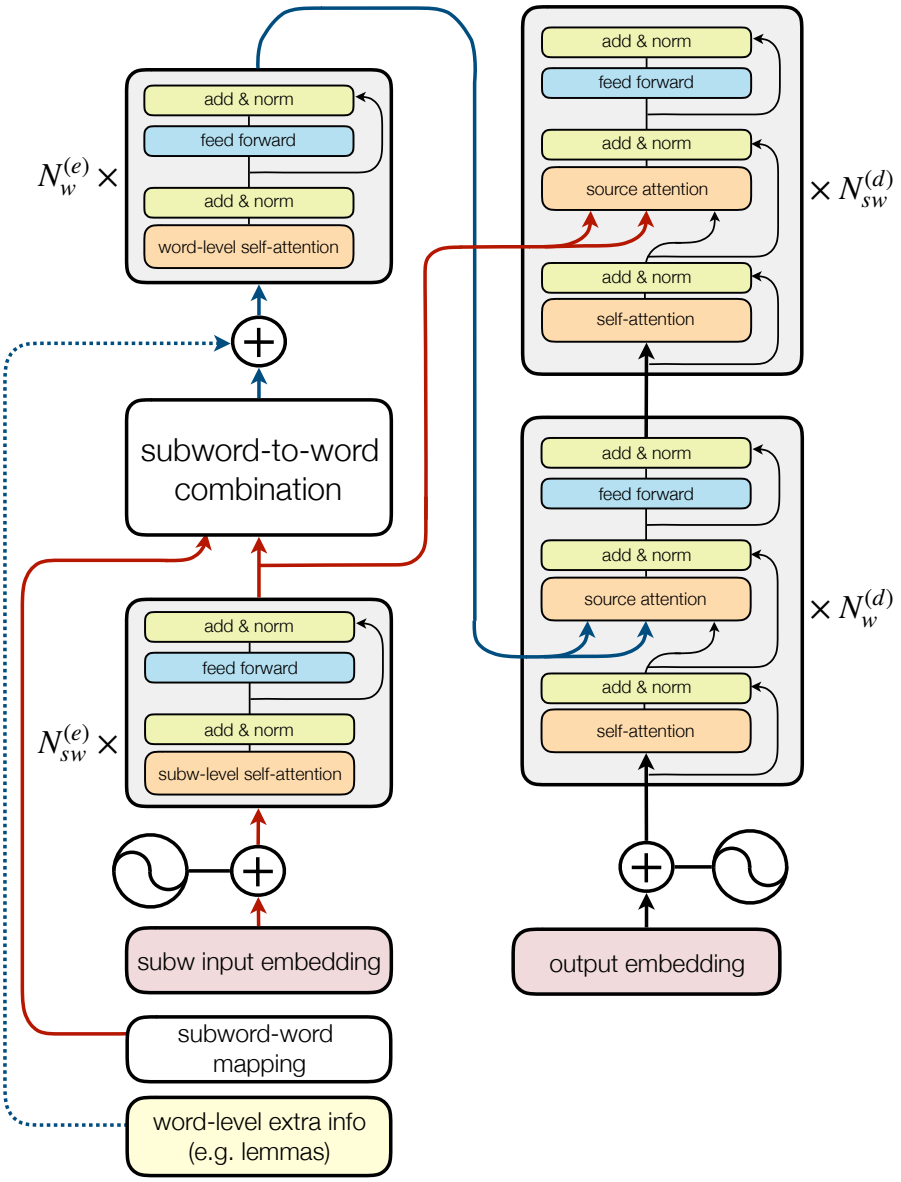

Figure 5.1.: Subword to Word Transformer model.

understand that such a configuration made it impossible for the network to properly handle token copying from source to target, which is usually needed in cases of proper nouns or compositional structures like numbers. Other possible causes for this degradation could be some mismatch on the encoder side e.g. positional embeddings being subword-based but encoder embeddings being word-level. To test this hypothesis, we added positional encodings after the point where subword representations are combined into word-level representations. This led to no improvement, suggesting that the inability to copy may be the cause of the degradation.

Several strategies were studied for the combination of subword representations into word representations , namely Long-Short Term Memories (LSTM; Hochre-

iter and Schmidhuber, 1997), Gated Recurrent Units (GRU; Cho et al., 2014) and simply adding all subwords within each word. In the case of LSTMs and GRUs, the inputs to the recurrent units are the subword representations, while only the outputs at the final subword position of each word are retained as the outputs of the combination block. In the case of the simple addition, the subword representations of each word get added together in a single vector. In all cases, the lengths of the sequences in the batch after the combination block is the number of word tokens in each sentence. After quantifying the effects of each strategy, the specific approach chosen to combine subword representations into word representations are GRUs.

The proposed approach provides a natural point to incorporate word-level information: after the subword-level representations have been combined into word-level ones. This way, as shown in Figure 5.1, the extra word-level information is embedded into a vector space and added to the word-level representations of the source sentence, after the word-to-subword combination.

5.2. Experimental Setup

We understand that there are two desirable properties for the proposed word-subword combination model: to be able to retain the translation quality obtained with the analogous subword-based model and to be able to better profit from word-level information than other approaches.

In order to verify that the translation quality is retained, we performed experiments on the IWSLT14 English-German data, both in English→German and German→English translation directions, with a shared subword vocabulary with 10K merge operations. We studied the resulting translation quality with different hyperparameter sets in order to understand their effect on the model.

In order to study the effectiveness of the proposed model with other approaches to incorporate word-level information into a subword-based model, we used the WMT16 English-Romanian data with the back-translated synthetic data from (Sennrich et al., 2016a), using a shared subword vocabulary of 40k merge operations.

We used the proposal by (Sennrich and Haddow, 2016) as baseline, and compared it to a vanilla Transformer baseline and to our proposed method.

For all experiments, we used the `fairseq` library (Ott et al., 2019), either with its built-in models for the baselines or with custom model implementations for

the approach by Sennrich and Haddow (2016) and for our own proposed architecture.

For the IWSLT14 de-en and en-de baselines we used the Transformer architecture (Vaswani et al., 2017) with the hyperparameters proposed by the `fairseq` authors[1], namely 6 layers in encoder and decoder, 4 attention heads, embedding size of 512 and 1024 for the feedforward expansion size, together with dropout of 0.3 and a total batch size of 4000 tokens, using label smoothing of 0.1. For the WMT16 en-ro baseline we used the base configuration of the Transformer model offered in `fairseq`, that is, 6 layers in encoder and decoder, 8 attention heads, embedding size of 512 and 2048 for the feedforward expansion size, together with dropout of 0.1 and total batch size of 32000 tokens, without label smoothing (following the baseline used by Gu et al. (2018a)). All reported BLEU scores are computed with the model weights averaged over the last 10 checkpoints after training until convergence.

5.3. Results

We studied the effect of different hyperparameter values over translation quality. We measured the results obtained on the IWSLT14 de-en data by using different types of subword combination strategies, as well as combining subwords at different layer levels, chosen arbitrarily. Table 5.1 shows how the subword combination strategy that obtains best results is to use GRU units that receive the subwords as input and return the outputs at the positions of the final subword in each word. The difference with the other alternatives is minimal, though. The rest of the hyperparameters are the same as the IWSLT14 baseline, with a total batch size of 12000 and the subword merging layers being $N_{sw}^{(e)} = 3$ and $N_{sw}^{(d)} = 3$.

Combination	**BLEU**
Addition	33.93
GRU	34.02
LSTM	33.92

Table 5.1.: BLEU scores on IWSLT14 German-English for different subword combination strategies.

Regarding the influence over the translation quality of the level at which subword representations are merged, Table 5.2 shows that the best results are obtained when merging subwords after the fifth encoder layer, and using again the subword representations in the decoder after the third layer. The rest of hyperparameters

[1] https://github.com/pytorch/fairseq/tree/master/examples/translation

are the same as the IWSLT14 baseline, with a total batch size of 12000 and GRU as subword combination strategy.

$N_{sw}^{(e)}$	$N_{sw}^{(d)}$	BLEU
3	5	33.53
3	3	34.02
5	3	34.46

Table 5.2.: BLEU scores on the IWSLT14 German-English test set for different values of $N_{sw}^{(e)}$ and $N_{sw}^{(d)}$, using GRU as subword combination strategy.

Once determined that using GRU as subword combination and setting $N_{sw}^{(e)} = 5$ and $N_{sw}^{(d)} = 3$ is the hyperparameter configuration that gives the best results, we checked whether the proposed architecture maintains the translation quality with respect to a vanilla Transformer baseline. As shown in Table 5.3, the BLEU scores are practically the same for both architectures and both German→English while for English→German there is a small decrease. As commented in section 5.2, the baseline uses a batch size of 4000 while our approach uses 12000. Note that for the baseline architecture, too large batch sizes actually decrease the resulting translation quality due small size of the training data; the value used is the standard one used for small training data sizes in `fairseq`[2]. The batch size for our architecture was chosen by manual fine tuning.

The encoder and decoder embeddings of the base transformer baselines were shared, with the final decoder projection being also tied to the embedding matrix. In the models with linguistic information, both the factored baselines and our models, given the differences between input and output vocabularies, we decided not to share the encoder and decoder embeddings, while the final projection of the decoder was tied to the input embedding matrix.

	en-de	de-en
Base Transformer	28.75	34.44
Word-subword model	28.29	34.46

Table 5.3.: BLEU scores on the IWSLT14 German-English data, using no extra word-level information.

Finally, in order to assess our proposed approach at incorporating extra word-level information, we compared it against the approach by Sennrich and Haddow (2016) (with the Transformer as base architecture), which copies the word level information to each of the subwords in the word; in our implementation, the

[2] https://github.com/pytorch/fairseq/tree/master/examples/translation

subword embedding and the linguistic information are combined by adding them together, which is analogous to the original alternative that concatenates them. For the vanilla Transformer and the approach by Sennrich and Haddow (2016) we used a total batch size of 32000 while for the word-subword model (our proposal), we used a total batch size of 40000, GRU as subword combination strategy and $N_{sw}^{(e)} = 5$ and $N_{sw}^{(d)} = 3$.

	en-ro
Base Transformer	27.02
Word-level info copied to subwords	27.29
Word-subword model + word-level info	27.82

Table 5.4.: BLEU scores measured on the WMT16 English-Romanian data, with lemmas as linguistic info.

The word-level linguistic information used was only the lemma (using a vocabulary of 40k lemmas), which is the feature that should provide the largest improvement according to Sennrich and Haddow (2016). We used Stanford CoreNLP (Manning et al., 2014) to annotate the corpus with the English lemmas. The obtained results are shown in Table 5.4, where our proposed approach obtains the best BLEU score compared to the base Transformer model (Vaswani et al., 2017) without any word-level information, and to copying the word-level info to subwords (Sennrich and Haddow, 2016).

5.4. Conclusion

In this chapter, we proposed a modification to the Transformer architecture to merge the subword representations from the first layers of the encoder into word-level representations. Merging word-level representations inside the model allows it to use the subword-level representations in the final decoder layers so that it can handle compositional structures and other situations where copying from source is needed. This approach provided an appropriate point to incorporate linguistic word-level information and it is superior at doing so compared with the reference approach by Sennrich and Haddow (2016).

Further work may include detailed characterization of the linguistical qualitative differences between the output of the baselines and our proposed approach to better diagnose the obtained quantitative results, as well as applying it to character-level instead of subword representations, and using it for morphologically richer languages, especially low-resourced agglutinative ones, where our

approach, together with the incorporation of linguistic information, may provide larger improvements in translation quality.

Further extensions may include studying the behavior of more powerful subword combination strategies and the application of subword merging to the target side. Note that applying this approach to the encoder part, as we do in this work, is straightforward, while applying the same approach to the decoder would present a key challenge: at inference time, the target side tokens are generated autoregressively one by one, which implies that it is not possible to combine all of the subword tokens of a word until they have all been generated.

6. Syntax-driven Iterative Expansion Language Models for Text Generation

The currently dominant text generation paradigm is based on generating a sequence of discrete tokens in a left-to-right autoregressive way. Most neural language models (LMs) fall into this autoregressive generation category. Some neural architectures are sequential in nature, such as those based on recurrent neural networks (RNNs), lending themselves naturally to the autoregressive approach when used together with teacher forcing Williams and Zipser (1989). Other architectures, such as Transformer Vaswani et al. (2017), while not intrinsically sequential, have also been targeted for sequential generation. On the other hand, some recent lines of research have focused on nonsequential generation.

In this chapter, we propose a new paradigm for text generation and language modeling called Iterative Expansion Language Model, which generates the final sequence following a token ordering defined by the sentence dependency parse by iteratively expanding each level of the tree. Related works regarding non-sequential language generation models is described in Section 2.3.4.

6.1. Iterative Expansion LMs

Our proposal is to train a new kind of language model where the token generation order is driven by the dependency parse tree of the sentence and where the generation process is iterative.

The input vocabulary contains terminal tokens as well as non-terminal special tokens called dependency placeholders, each of which is associated with one of the possible dependency relations to the heads. For the dependency tree in Figure 6.1, the dependency placeholders are `[poss]`, `[nsubj]`, `[advmod]`, `[xcomp]`, `[dobj]` and `[ROOT]`.

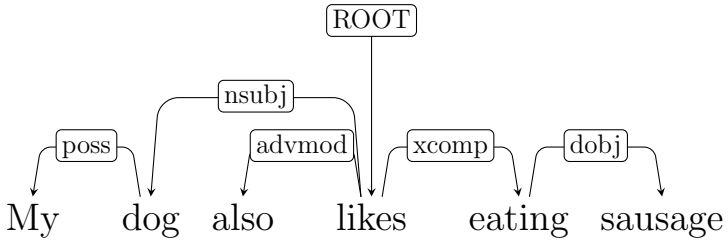

Figure 6.1.: Example of dependency parse tree.

The input of the first iteration is the sequence with the `[ROOT]` element. At each iteration, the model receives as input a sequence I_{tok} with tokens from the input vocabulary and non-autoregressively generates two new sequences, each with the same length as the input.

The first output sequence, O_{tok} , contains tokens from a vocabulary with all possible textual tokens (terminal tokens). The second output, O_{exp} , is a sequence of tokens called expansion placeholders, which are taken from a separate vocabulary. Each expansion placeholder is associated with a pattern describing the left and right dependencies of the token at that position in the O_{tok} sequence. An example of dependency expansion could be `[nsubj-advmod-HEAD-xcomp]` for the word "likes" in the dependency parse tree from Figure 6.1.

After each iteration, the output of the model is expanded.[1] This consists of creating a new sequence by combining the tokens from I_{tok} , O_{tok} and O_{exp} . This process is illustrated in Figure 6.2, making use of the dependency tree from Figure 6.1.

When there is a padding token `[pad]` in the output (either O_{tok} or O_{exp}), this means that the output at that position is ignored when computing the loss function. This occurs when the terminal token has already been computed in previous iterations and has therefore been received as part of I_{tok} , and the model does not need to compute it again.

Note also that an empty dependencies token `[HEAD]` marks the end of a branch and that there is no need for an end of sequence token `<eos>`. As shown in the example from Figure 6.1, the generation of independent branches occurs in parallel, needing only 3 iterations to generate a 6-token sentence.

The strategy for composing tree expansion tokens (e.g., `[nsubj-advmod-HEAD-xcomp]`) may not scale well when single words have many direct dependencies. To alleviate this, we introduce a preprocessing step to modify the dependency tree so

[1] The expansion of the output to be fed as input in the next iteration occurs in the CPU outside of the neural model itself.

Iteration 1						
I_{tok} :			[ROOT]			
O_{tok} :			likes			
O_{exp} :	[nsubj-advmod-HEAD-xcomp]					

Iteration 2				
I_{tok} :	[nsubj]	[advmod]	likes	[xcomp]
O_{tok} :	dog	also	[pad]	eating
O_{exp} :	[poss-HEAD]	[HEAD]	[pad]	[HEAD-dobj]

Iteration 3						
I_{tok} :	[poss]	dog	also	likes	eating	[dobj]
O_{tok} :	my	[pad]	[pad]	[pad]	[pad]	sausage
O_{exp} :	[HEAD]	[pad]	[pad]	[pad]	[pad]	[HEAD]

Figure 6.2.: Example of iterative text generation.

that every word has at most one dependency to the left and one to the right. For each word with more than one dependency on any of its sides, we rearrange the tree to force left-to-right dependencies. Although this **tree binarization** reduces the degree of parallelism, it reduces data sparsity and allows handling constructions with a number of dependencies may otherwise be too large for the model to properly capture, such as enumerations (e.g., "I bought a pair of shoes, an umbrella, a beautiful jacket and a bracelet").

Iterative expansion LMs can be naturally extended to subword vocabularies, like byte-pair encoding (BPE; Sennrich et al., 2016c): for each word, we decompose its node in the tree into as many nodes as subwords in the word, rearranging the tree so that the head of the old word is now the head of the first subword, and each subsequent subword depends on the previous one, while every dependency of the old word node now depends on the last subword.

6.1.1. Neural Architecture

The neural architecture proposed is based on a Transformer decoder Vaswani et al. (2017). To generate the dual output (terminal tokens and expansion placeholders) we condition the generation of terminals on the expansions: the probability distribution over the expansion token space is generated first by projecting from one of the intermediate layers' hidden states. We sample from it and use the resulting expansion IDs as an index to a trainable expansion embedding layer;

the embedded vectors are added to the hidden state used to generate them for use as input to subsequent layers.

As described in Section 6.1, the input and output token vocabularies are different: the latter only contains terminal tokens (plus some special tokens such as `[PAD]`); the former also contains dependency placeholders. However, for practical purposes, at the model level, we define both vocabularies to be the same, both with terminal tokens and dependency placeholders, and we mask the entries of dependency placeholders in the final softmax.

To inject the syntactic dependency information as input into the model, we add a layer of learned positional embeddings containing the position of the head of each token, and we refer to this embedding layer as head position embedding.

The self-attention mask used in Transformer to force causality is not used in our proposal. The input is therefore not masked at all, and the token predictions have access to the full input sequence.

6.1.2. Training

For training iterative expansion LMs, the main input of the model is the tokens at one of the levels of the dependency parse tree (I_{tok}), while the output is the following level tokens (O_{tok}) and expansion placeholders (O_{exp}). Secondary inputs to the model are the dependency indexes (which are used in the head position embedding) and the mask used for the constrained attention variant.

The model is trained with maximum likelihood on the categorical cross-entropy for both tokens and expansion placeholders, then adding both sublosses into the final loss. Tokens generated in previous iterations appear as `[PAD]` tokens in the expected output and are ignored when computing the loss.

Training takes place in batches; as the trainable unit is a level transition, a training batch is composed of level transitions from different sentences.

6.1.3. Inference and Text Generation

In iterative expansion LMs, inference takes place iteratively. The initial state is a batch of `[ROOT]` tokens, together with the head positions initialized to the special value representing the root node and, in constrained attention variants, a mask with the self-dependency of the single node in each sentence in the batch. At each iteration, the model generates the probability distributions for terminal tokens and expansion tokens. We use nucleus sampling (Holtzman et al., 2020)

to sample from them. The terminal token sequences are expanded according to the expansion tokens (see Section 6.1), and these are the inputs for the following iteration if there are still unfinished branches. Before sampling from the token and expansion probability distributions, we mask the `<unk>` token and the dependency placeholders to avoid generating them.

6.2. Experimental Setup

6.2.1. Unconditional Text Generation

We conducted experiments on unconditional text generation following the methodology used by Caccia et al. (2020). The goal is to assess both the quality and diversity of the text generated by the model and the baselines. For the quality evaluation, we use the BLEU score Papineni et al. (2002) over the test set, where each generated sentence is evaluated against the whole test set as a reference. For diversity, we used the self-BLEU score Zhu et al. (2018), computed using as references the rest of the generated sentences. For each model, the temperature of the final softmax τ is tuned to generate text in the closest quality/diversity regime to the training data.

Iterative expansion LMs are compared against a standard LM baselines, namely, AWD-LSTM[2] Merity et al. (2018) and a Transformer LM Vaswani et al. (2017), both with word (w) and BPE subword (sw) vocabularies. The models were trained on the EMNLP2017 News dataset enriched with dependency annotations by `corenlp`. Syntax-driven generation baseline models were not included because the only model with an available implementation that is able to do unsupervised text generation are RNNGs, but they proved not to scale even to medium-sized datasets like EMNLP2017 News. When sampling from models, we use nucleus sampling Holtzman et al. (2020), a form of ancestral sampling that constrains the candidate pool by discarding the distribution tail. Samples from the training and validation data are included for reference.

6.2.2. Style Variation

Iterative expansion LMs drive the generation of text with the dependency parse tree. It is possible to influence the generated trees by altering artificially the probability of the different expansion tokens. To demonstrate this, we modified

[2]Abbreviation of ASGD weight-dropped LSTM, where ASGD stands for averaged stochastic gradient descent.

the decoding process of iterative expansion LMs to force the probability of generating adjectival constructions to be higher than normal, aiming at generating a more descriptive style: during decoding, we multiply the probabilities of the expansion placeholders that express adjectival dependencies (i.e. those containing adjectival modifier "amod" relations), and renormalize the probabilities by dividing by the sum.

We conducted this experiment with the word-level models trained on EMNLP2017 News data. We compute the ratio of adjectives per sentence to verify the increased presence of adjectives, while controlling quality and diversity measures over the generated text for potential degradation.

6.3. Datasets and Preprocessing

In this section we present the datasets used for our experiments, including the relevant statistical figures, together with the preprocessing steps applied to the data.

6.3.1. Dataset Statistics

Table 6.1 summarizes the statistics of the EMNLP2017 News dataset used in our experiments. The training/validation/test split was taken from the work by Holtzman et al. (2020).[3]

	train	**valid**	**test**
sentences	268k	10k	10k
iterations	3.2M	122k	122k
expansion vocab	904		
terminal vocab	8195		

Table 6.1.: Statistics of the EMNLP2017 News dataset.

6.3.2. Data Processing Details

The tokenization of the EMNLP2017 News dataset is very nonstandard. To appropriately prepare it to be used as input to the syntactic annotation tool

[3]The EMNLP 2017 News data can be downloaded from https://github.com/pclucas14/GansFallingShort/tree/master/real_data_experiments/data/news

`corenlp`, we detokenized the text and then retokenized it again with the Moses tokenizer. For the experiments with BPE, we created the subword vocabulary with 4000 merge operations and without further constraining the size of the resulting vocabulary.

Text generation with AWD-LSTM. AWD-LSTM is trained with "continuous" text batches. This implies that when used for text generation, it likewise generates text. To obtain a predetermined number of sentences, we used AWD-LSTM to generate a fixed number of tokens (e.g., 200). Then, we split this text at the `<eos>` boundaries and removed the first and last sentences to avoid incomplete ones. We repeated this procedure until we had the target number of sentences.

Text generation with the Transformer. A Transformer LM was trained following the data preparation instructions in the fairseq examples.[4]

Quality vs. diversity plots. The generated text was un-BPE'ed (for the subword-level models) and detokenized by means of the Moses `detokenizer.perl` script. Then, it was tokenized with the Moses `tokenizer.perl` script, and the BLEU scores were computed with the NLTK `corpus_bleu` function Loper and Bird (2002), without smoothing.

GPT-2 perplexity computation. The text that served as input to GPT-2 was properly detokenized before applying the model's own BPE tokenization.

6.4. Hyperparameter Configuration

In this section, we present the detailed hyperparameters used in our experiments. They were obtained by manual exploration, observing the behavior of the loss over the training and validation sets of each dataset. The number of manual hyperparameter search trials were less than 10 for each model.

The hyperparameters of the iterative expansion LM models used for the text generation experiments presented in Figure 6.3, for both the word and subword vocabulary variants, are shown in Table 6.2.

The hyperparameters of the AWD-LSTM baseline are presented in Table 6.3. Note that the AWD-LSTM variant used as a baseline is the base LM without the continuous cache pointer mechanism, with tied weights. Additionally, note that the terminal and expansion vocabulary sizes are different, which leads to a different size of the expansion embedding table and therefore to a different total number of parameters for the same values of the rest of the hyperparameters.

[4] https://github.com/pytorch/fairseq/tree/master/examples/language_model

num. layers	6
num. heads	8
embed. size	1024
batch size	16384
num. params	96M

Table 6.2.: Hyperparameters of the iterative expansion LM used in the text generation experiments.

hidden size	1150
embed. size	400
num. layers	3
batch size	20
BPTT	70
num. params	23.5M

Table 6.3.: AWD-LSTM baseline hyperparameters.

The hyperparameters of the Transformer baseline are presented in Table 6.4. We used the implementation of the fairseq library and tuned it on the training and validation data.

num. layers	6
num. heads	4
embed. size	512
batch size	16384
num. params	17M

Table 6.4.: Transformer baseline hyperparameters.

The batch size for ItExp is expressed in total number of tokens, while for AWD-LSTM it is expressed as number of sentences, which, when multiplied by the back-propagation through time (BPTT) length, gives the total number of tokens per batch. Note that the criteria for the optimum batch size differ for transformers and LSTMs.

Note that the hyperparameters of each model are tuned separately, independently from the other models, leading to differences in the total number of parameters. While this makes the models less comparable, just comparing models with similar number of parameters would lead to artificially better or worse performance for a specific dataset (characterized by its size, its word distribution, etc). Instead of that, we chose for each model the best hyperparameter configuration for a specific dataset. This, allows better understanding of the optimal characteristics

of each model for a specific dataset profile and unveils the "parameter efficiency" of each model.

To sample from both our proposed model and the baselines, we use nucleus sampling Holtzman et al. (2020) with $p = 0.9$.

6.5. Results and Analysis

We assess the ability of iterative expansion LMs to unconditionally generate text in terms quality (BLEU-5) vs. diversity (self BLEU-5), comparing against sequential baselines, each with a softmax temperature τ tuned separately.

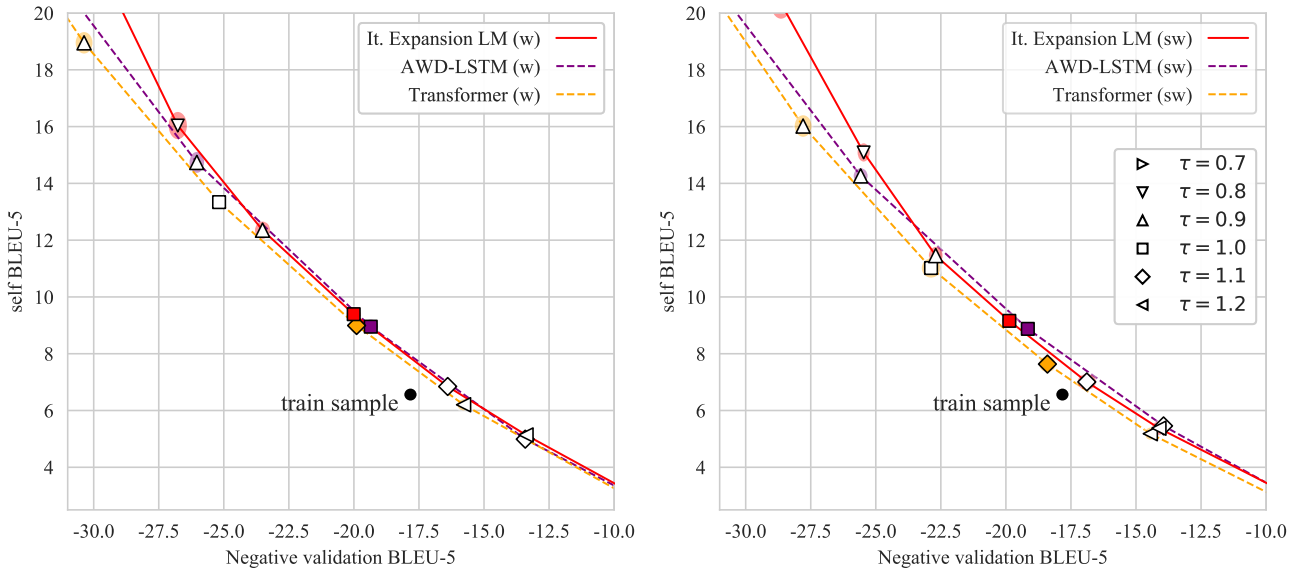

Figure 6.3.: Quality vs. diversity on EMNLP2017 News (BLEU-5). Models with **word-level vocabulary on the left** and **subword-level on the right**. The point marker is color-filled for the chosen value of τ. Each point represents the average over 20 generated text samples, and is surrounded by a small colored ellipse representing the standard deviation.

In order to tune the output softmax termperature τ, we generated text with each model at different temperatures and chose the value of τ that was the most similar to a sample from the training data in terms of BLEU-5 against a sample from the validation set (proxy for quality) and self BLEU-5 (proxy for diversity). Each model was used to generate 20 samples of 400 sentences, and self-BLEU5 and validation-BLEU5 were computed over each of them, taking the average and the standard deviation. Figure 6.3 and Table 6.5 show these BLEU values, highlighting the chosen τ for each model. Given the low values for the standard

τ	ItExp (w)		AWD-LSTM (w)		Transformer (w)	
	valid ↑	self ↓	valid ↑	self ↓	valid ↑	self ↓
0.70	30.1 ± 0.8	22.3 ± 1.0	39.2 ± 0.9	33.4 ± 1.1	40.5 ± 0.6	35.0 ± 1.1
0.80	26.8 ± 0.8	16.0 ± 1.0	33.0 ± 0.7	23.2 ± 1.0	35.8 ± 0.7	26.3 ± 0.8
0.90	23.5 ± 0.7	12.4 ± 0.7	26.0 ± 0.6	14.7 ± 0.8	30.4 ± 0.7	19.0 ± 0.8
1.00	$\mathbf{20.0 \pm 0.6}$	$\mathbf{9.4 \pm 0.5}$	$\mathbf{19.4 \pm 0.6}$	$\mathbf{9.0 \pm 0.6}$	25.2 ± 0.5	13.3 ± 0.5
1.10	16.4 ± 0.5	6.8 ± 0.5	13.4 ± 0.4	5.0 ± 0.4	$\mathbf{19.9 \pm 0.6}$	$\mathbf{9.0 \pm 0.6}$
1.20	13.4 ± 0.6	5.1 ± 0.4	9.0 ± 0.5	2.9 ± 0.3	15.8 ± 0.5	6.2 ± 0.5

τ	ItExp (sw)		AWD-LSTM (sw)		Transformer (sw)	
	valid ↑	self ↓	valid ↑	self ↓	valid ↑	self ↓
0.70	28.6 ± 0.9	20.3 ± 1.1	39.0 ± 0.8	33.5 ± 1.1	36.9 ± 0.7	30.6 ± 1.2
0.80	25.5 ± 0.5	15.1 ± 0.7	32.3 ± 0.7	22.4 ± 0.7	32.5 ± 0.7	22.4 ± 1.0
0.90	22.7 ± 0.6	11.5 ± 0.7	25.6 ± 0.6	14.3 ± 0.6	27.8 ± 0.7	16.0 ± 0.8
1.00	$\mathbf{19.9 \pm 0.6}$	$\mathbf{9.2 \pm 0.5}$	$\mathbf{19.2 \pm 0.5}$	$\mathbf{8.9 \pm 0.5}$	22.9 ± 0.8	11.0 ± 0.7
1.10	16.9 ± 0.8	7.0 ± 0.6	13.9 ± 0.5	5.5 ± 0.4	$\mathbf{18.4 \pm 0.7}$	$\mathbf{7.6 \pm 0.6}$
1.20	14.1 ± 0.6	5.4 ± 0.5	9.7 ± 0.4	3.3 ± 0.3	14.5 ± 0.5	5.2 ± 0.5

Table 6.5.: Validation and self BLEU-5 scores of the text generated by the **word-level (top)** and **subword-level (bottom)** models under study at different temperatures τ, showing the average and standard deviation over 20 different generated text samples. The selected generation regime is highlighted for each model, being the closest to the training sample, which has a validation BLEU-5 of 17.8 and a self BLEU-5 of 6.6.

deviation, we decided not to include it in subsequent tables to avoid unnecessary clutter. Note that in all BLEU vs. self-BLEU figures, each model is shown as a different line (each with its own color and/or dashed pattern) and that the data points computed for each temperature value are plotted with a specific marker shape (square, diamond, triangle, or flipped triangle). We can appreciate that the temperature regimes affect AWD-LSTM and iterative expansion LMs differently, with the latter concentrating around the training/validation sample points.

Apart from BLEU scores, we also include extra quality measures, namely the perplexity obtained by other language models: an AWD-LSTM word-level LM and a Transformer word-level LM, both trained on EMNLP2017 News, plus OpenAI GPT-2 (1.5 B parameters) Radford et al. (2019). The results are shown in Table 6.6.

These results show how the generated text improves over AWD-LSTM in terms of quality by all measures, with a comparable level of diversity. In comparison to the Transformer, while the quality measured with BLEU-5 is better for ItExp, the

	τ	Test BLEU-5 (quality ↑)	Self BLEU-5 (diversity ↓)	AWD-LSTM perplex. ↓	Transformer perplex. ↓	GPT-2 perplex. ↓
AWD-LSTM (w)	1.0	22.9	8.9	37.0	47.9	99.5
Transformer (w)	1.1	23.8	9.0	33.6	18.6	66.5
ItExp (w)	1.0	23.7	9.4	40.8	40.7	85.2
AWD-LSTM (sw)	1.0	22.7	8.9	43.5	56.9	113.5
Transformer (sw)	1.1	22.1	7.6	37.5	31.6	77.1
ItExp (sw)	1.0	23.6	9.2	45.2	49.2	97.1
Train sample	-	21.5	6.6	49.5	29.1	37.7
Valid sample	-	21.2	7.2	53.3	44.7	36.7

Table 6.6.: Quality and diversity on EMNLP2017, with τ generating the closest text to the validation data.

rest of the quality measures indicate that the text generated by the Transformer is of better quality.

Adjective probability	Adjs. per sentence	Test BLEU-5	Self BLEU-5
×1	1.2	23.7	9.4
×10	3.4	21.3	8.4
×20	4.2	20.6	8.8
×50	5.2	19.8	8.9

Table 6.7.: ItExp (w, $\tau = 1.0$) with increased adjectives.

The results of the styled text generation experiments, shown in Table 6.7, confirm that the style of the resulting text can be successfully modulated to the desired degree and that the quality and diversity are only slightly degraded at moderate increases of the probability of adjectival clause generation.

6.5.1. Human Evaluation

In order to better assess the quality of the generated text, we also include a human evaluation. For this, we took a sample of 60 sentences of each model under study, including also a sample of the same size from the validation data, to serve as reference. The sentences were evaluated by a pool of annotators, who were requested to rate the sentence in an integer scale from 1 to 5, taking into account its fluency and correctness.

The pack of sentences rated by each annotator contained 10 sentences from each of the models under evaluation. Each sentence under evaluation was part of the packs of 3 evaluators; this redundancy was used to measure the discrepancies in the rating of each sentence among annotators, which was quantified by means of the average per-sentence standard deviation.

Model	Average rating	Per sentence avg. stddev
AWD-LSTM (w)	3.08	0.74
Transformer (w)	3.43	0.78
ItExp (w)	3.28	0.73
AWD-LSTM (sw)	2.66	0.68
Transformer (sw)	3.33	0.83
ItExp (sw)	3.09	0.70
Valid sample	4.49	0.61

Table 6.8.: Human evaluation for the different models.

Table 6.8 shows the statistics of the obtained ratings, were we can see the average rating of the sentences generated by each model, together with the average per-sentence standard deviation, to understand how different the ratings for each sentence were among the different evaluator ratings. We can see that the highest human ratings were obtained by the Transformer, both with word and subword-level vocabularies, followed by ItExp and then AWD-LSTM.

Adjective probability	Average rating	Per sentence avg. stddev
×1	3.28	0.73
×10	3.16	0.79
×20	2.98	0.84
×50	3.19	0.70

Table 6.9.: Human evaluation for ItExp (w) models with increased adjectival construction probability.

Table 6.9 shows the human evaluation for the models from the style variation experiments presented in Table 6.7. As we can see, there is a small degradation in quality as we force high levels of adjectival presence.

6.6. Further Comparison with Real Text

Given that the generation process in iterative expansion LMs is not sequential, we studied the distribution of the sentence lengths it generates. This is shown in Figure 6.4 for the text generated by a word-level iterative expansion LM trained on EMNLP2017 News, along with the lengths of a sample from the training data. We can appreciate that the sentence length distribution of iterative expansion LMs is very similar to the distribution of real text.

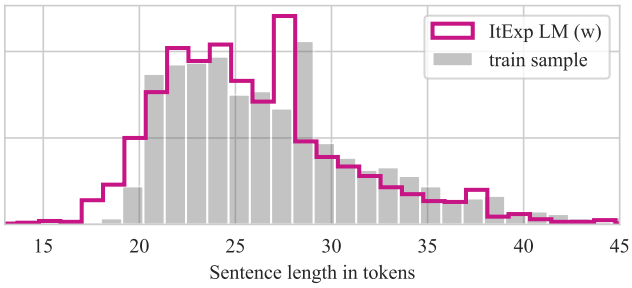

Figure 6.4.: Distribution of generated text length.

Iterative expansion LMs generate the dependency parse tree as they generate text. We studied the depths of the dependency trees of generated text in relation to those parsed from the training data, as shown in Figure 6.5. We can appreciate that the dependency tree depth distribution of iterative expansion LMs is remarkably similar to the distribution of real text.

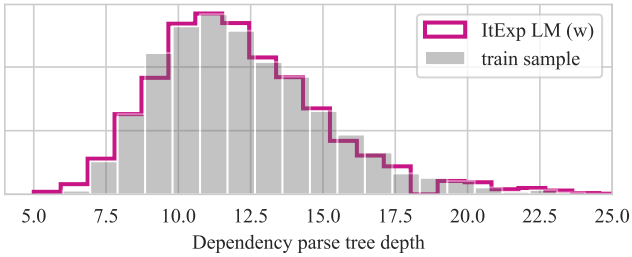

Figure 6.5.: Histogram of generated text tree depth.

We also measured the degree to which the generated trees adhere to the trees obtained by parsing their lexicalized representation. Specifically, we computed the labeled and unlabeled attachment scores between both for the text generated at different softmax temperatures τ.

τ	0.7	0.8	0.9	1.0	1.2
LAS	96.4	95.3	94.2	92.3	86.2
UAS	98.0	97.3	96.5	95.2	90.7

Table 6.10.: Attachment scores of the generated trees.

Attachment scores are the standard performance measure in dependency parsing and are computed as the percentage of words that have been assigned the same head as the reference tree, over a test set. The attachment score is "labeled" if the dependency label is taken into account or "unlabeled" otherwise. As shown in Table 6.10, the obtained labeled attachment scores (LAS) and unlabeled attachment scores (UAS) are very high across the different values of the generation temperature τ.

6.6.1. Quantification of the Generation Speedup

Text generation with autoregressive models like LSTM or Transformer models offers a linear computational complexity with respect to the length of the generated sequence. In comparison, the dependency tree-driven decoding used by iterative expansion LMs generates text in parallel for each branch in the tree. If the tree was a perfectly balanced binary tree, then the computational complexity would be logarithmic. However, dependency trees in general are not balanced and, given the tree binarization postprocessing that we introduce, the parallelization is slightly reduced.

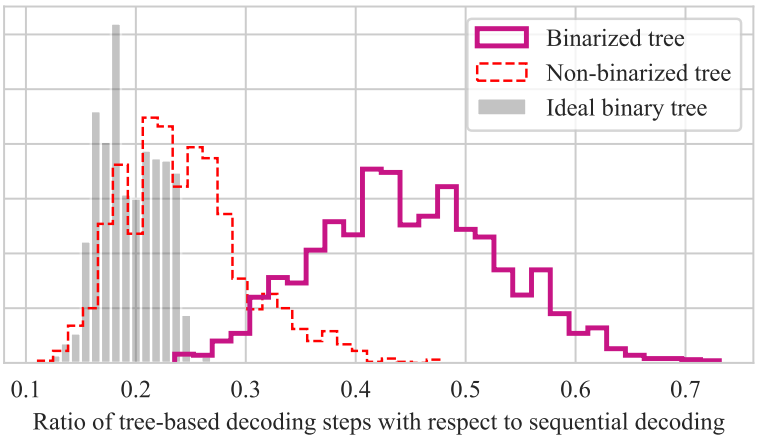

Figure 6.6.: Histogram of the ratio of the decoding steps needed to generate a sentence with tree-based decoding with respect to sequential generation.

Figure 6.6 shows the speedup of the needed decoding steps of tree-based decoding with respect of auto-regressive decoding, taking a sample of the training data and computing the needed steps to decode them should the sentences have an idealized binary dependency parse tree, a normal parse tree, and a binarized parse tree. On average, the binarized parse tree, which is the decoding used by iterative expansion LMS, needs only 45% of the decoding steps needed by autoregressive decoding.

Note that, while the computational complexity is improved with respect to autoregressive approaches, this improvement may not translate directly into execution speed. Some other factors to be taken into account in that regard are the penalty imposed for having two projection and softmaxes (one for the expansion tokens and one for the terminal tokens) instead of one, and the increasing length of the batch computations as the sentence is decoded instead of the almost constant decoding for autoregressive approaches, where the previous steps' computations are cached.

6.6.2. Generation Examples

Table 6.11 shows a selection of text samples generated by iterative expansion LMs with a word-level vocabulary, while Table 6.12 shows samples generated with a subword-level vocabulary. We can see that, despite being generated non-sequentially and each branch of the dependency parse tree being generated in parallel, the resulting sentences maintain coherence and syntactic agreement, confirming that conditioning on the token dependencies in the parse tree provides enough information to generate it while speeding up the decoding process.

American students were 62 percent more likely to die in a heart attack during the first week of 2004, according to the study.
For 150 days, Hillary Clinton will do more to improve access to affordable quality care, support and education funding for millions of Americans, she says.
For those on this list, it's likely that I would rather be able to train them up, she said.
He made it clear the SNP repeated on Friday as a response, saying they discussed a contract getting the extra cost here.
He'll pay $25, 000 for rent and more buses and bring his collection to The Academy on Channel 31.
Six years later, at least eight people died as a result of the shooting.
The health prime minister told CNN Thursday that he was willing to back up against the US and remove all of the relevant items at the end of the transition.
Then, another man told police that was a friend's friend, and as a child, he made the decision to call his mother.
They are 40 - 60 among the top 50, 000 women in the last year in that group since 2014 - 15.
They've worked hard on Twitter and they think they've tried to focus on our sport, she said.
We like to think that if you try to get this game done, we can get a lower success rate out of 15.

Table 6.11.: Samples of text generated by iterative expansion LMs with word vocabulary.

Finally, Figure 6.7 shows examples of generated sentences together with their dependency trees.

I feel that they're going to Syria because we had this explanation, that they have an indication of their advance.
The girl's mother told the group of three she needed treatment and the family said her daughter would still be alive with another child.
But she added: "The data is important to the EU that the UK can attract more businesses.
Though he also spoke to Mr Wilson on Saturday morning at the Netherlands Police trial, Johnson referred it to the No. 1 commission.
It's a collective belief and it's a statement to us, he said.
It's just the first thing we're feeling now and I don't like it.
So if you want to be sitting in a garden, you have to wait for something to make sure that this does not end.
So, for example, we need to argue about what the president did, but I'm just interested in having any talk.
The British defence ministry confirmed action had been taken at the hospital but could not confirm the details until now.
We'll ask for a fair share of Russia to stop border security, particularly for people of color, he added.

Table 6.12.: Samples of text generated by iterative expansion LMs with subword vocabulary.

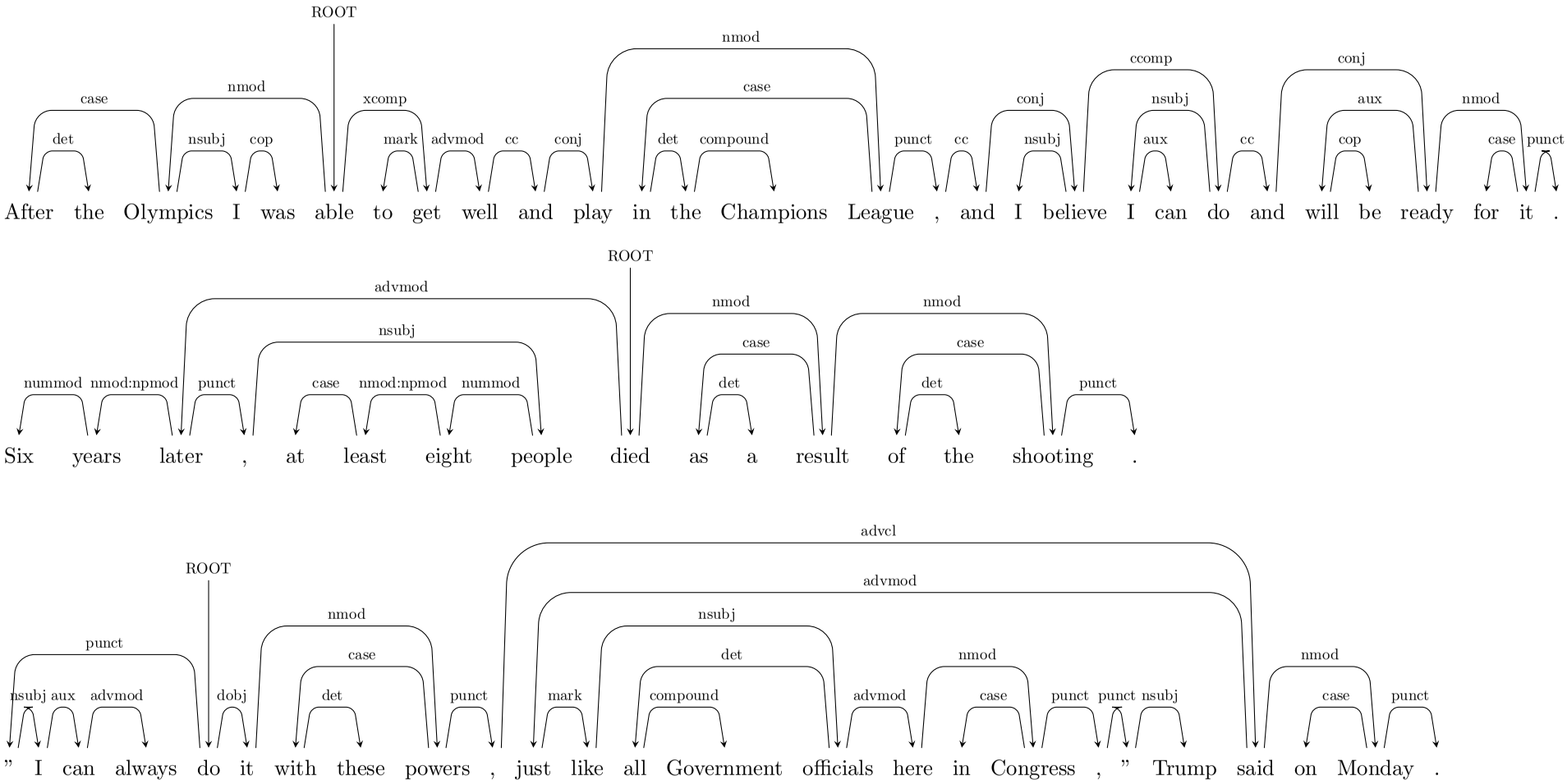

Figure 6.7.: Examples of generated sentences, together with their generated parse trees.

6.7. A Note on Perplexity Computation

Language models, apart from being used for text generation, can usually estimate the probability of a given sentence, or alternative its perplexity (see Section 2.3.3). Iterative Expansion LM can also be used for this but, like other syntax-driven LMs described in Section 2.5.3, can only reliably compute the perplexity of a sentence given a specific dependency parse tree of the sentence. While it is possible to approximate the unconditional perplexity by introducing some assumptions, our experiments showed that such approximations where not comparable with sequential language models.

6.8. Conclusion

We presented iterative expansion LMs, which are iterative non-autoregressive text generation models that rely on syntactic dependency trees to generate sentence tokens in parallel. As opposed to other syntax-driven generation mechanisms, the training of iterative expansion LMs can be naturally computed in batches and they are amenable to subword-level vocabularies.

We showed that our proposed method generates text with quality between LSTMs and Transformers, with comparable diversity, both regarding automatic measurements and human judgement, while generating text in half of the decoding steps needed by sequential LMs, and also allowing direct control over the generation process at the syntactic level, enabling the induction of stylistic variations in the generated text.

7. Conclusion

The use of linguistic information in NLP systems has been a recurrent line of research over time. However, with the dominance of NMT systems and the recent abundance of training data, the edge once provided by linguistic information has diminished, as training with more data can provide the same improvements. This, however, only holds applicable for the cases where there is plenty of data to train with. This is not the case of most of the languages in the world, which are low-resourced. It does not apply either to domains where the amount of available data is scarce. These are the cases where the use of linguistic information can make an impact. As described in Chapters 3 and 4, the use of morphological information can provide large improvements in translation quality of morphologically-rich languages in out-of-domain texts.

The specific approach to inject linguistic information into NMT systems is still an open problem, especially given the mismatch of the subword vocabularies normally used in NLP and the word-level granularity of the linguistic information. This mismatch also applies to other types of information that are defined at word-level, like semantic annotations. Chapter 5 provides an initial attempt at opening NMT systems to incorporating these types of input information.

Regardless of whether it can deliver improvements in the results in a specific data setup, the injection of linguistic information can play another important role: bridging neural networks and human understanding. Linguistic information is not only a characterization of text, but a tool to understand its structure. In Chapter 6, we force these linguistic structures into the very text generation process of a neural network. The result is that the generation is understandable by a human, as it is driven by syntactic constructions. Given this transparency, the generation process can even be controlled externally, to influence the generated text. This makes the linguistic information a bridge for the human to interact with the inner mechanisms of the network, forcing a small breach on the black box.

From this thesis, I have come to the conclusion that the injection of linguistic knowledge with the purpose of improving the results quality is still a useful

resource in some scenarios, but that it will play a more important part in the future devisal of interpretable and controllable neural systems that improve human understanding of the underlying system dynamics.

Bibliography

N. Akoury, K. Krishna, and M. 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/.

A. Alexandrescu and K. Kirchhoff. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 1–4, New York City, USA, June 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N06-2001.

J. A. Alonso and G. Thurmair. The comprendium translator system. In *Proceedings of the Ninth Machine Translation Summit*, 2003. URL http://mt-archive.info/MTS-2003-TOC.htm.

J. Armengol-Estapé, M. R. Costa-jussà, and C. Escolano. Enriching the transformer with linguistic and semantic factors for low-resource machine translation. *arXiv preprint arXiv:2004.08053*, 2020. URL https://arxiv.org/abs/2004.08053.

M. Artetxe, G. Labaka, and E. Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1250. URL https://www.aclweb.org/anthology/D16-1250.

M. Artetxe, G. Labaka, and E. Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1042. URL https://www.aclweb.org/anthology/P17-1042.

M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sy2ogebAW.

M. Artetxe, G. Labaka, N. Casas , and E. Agirre. Do all roads lead to Rome? Understanding the role of initialization in iterative back-translation. *Knowledge-Based Systems*, page 106401, 2020. ISSN 0950-7051. doi: 10.1016/j.knosys.2020.106401. URL http://www.sciencedirect.com/science/article/pii/S0950705120305335.

S. Assem and S. Aida. Machine translation of different systemic languages using a apertium platform (with an example of English and Kazakh languages). In *2013 International Conference on Computer Applications Technology (ICCAT)*, pages 1–4, Jan 2013. doi: 10.1109/ICCAT.2013.6522002. URL https://doi.org/10.1109/ICCAT.2013.6522002.

D. Ataman and M. Federico. Compositional representation of morphologically-rich input for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 305–311, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2049. URL https://www.aclweb.org/anthology/P18-2049.

D. Ataman, M. Negri, M. Turchi, and M. Federico. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331 – 342, 2017. doi: https://doi.org/10.1515/pralin-2017-0031. URL https://content.sciendo.com/view/journals/pralin/108/1/article-p331.xml.

B. Babych and T. Hartley. Extending the BLEU MT evaluation method with frequency weightings. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 621–628, Barcelona, Spain, July 2004. doi: 10.3115/1218955.1219034. URL https://www.aclweb.org/anthology/P04-1079.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of*

the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W05-0909.

C. Basta, M. R. Costa-jussà, and N. Casas . Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39. Association for Computational Linguistics, Florence, Italy, Aug. 2019. doi: 10.18653/v1/W19-3805. URL https://www.aclweb.org/anthology/W19-3805.

C. Basta, M. R. Costa-jussà, and N. Casas . Extensive study on the underlying gender bias in contextualized word embeddings. *Neural Computing and Applications*, 2020. URL https://doi.org/10.1007/s00521-020-05211-z.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. URL https://ieeexplore.ieee.org/document/279181.

Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 932–938. MIT Press, 2001. URL http://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003. URL http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X. URL https://www.aclweb.org/anthology/Q17-1010/.

O. Bojar, C. Federmann, B. Haddow, P. Koehn, M. Post, and L. Specia. Ten years of wmt evaluation campaigns: Lessons learnt. In *Proceedings of the LREC 2016 Workshop Translation Evaluation – From Fragmented Tools and Data Sets to an Integrated Ecosystem*, 2016. URL http://www.lrec-conf.org/proceedings/lrec2016/workshops/LREC2016Workshop-MT%20Evaluation_Proceedings.pdf.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993. URL https://www.aclweb.org/anthology/J93-2003.

J. Buys and P. Blunsom. Neural syntactic generative models with exact marginalization. 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 942–952, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1086. URL https://aclweb.org/anthology/N18-1086/.

M. Caccia, L. Caccia, W. Fedus, H. Larochelle, J. Pineau, and L. Charlin. Language gans falling short. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgza6VtPB.

C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, Apr. 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E06-1032.

M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico. Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, volume 57, 2014. URL http://www.mt-archive.info/10/IWSLT-2014-Cettolo.pdf.

W. Chan, N. Kitaev, K. Guu, M. Stern, and J. Uszkoreit. KERMIT: Generative insertion-based modeling for sequences. *arXiv preprint arXiv:1906.01604*, 2019. URL https://arxiv.org/abs/1906.01604.

C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, and D. Wu. Structure and performance of a dependency language model. In *In Proceedings of Eurospeech*, pages 2775–2778, 1997. URL https://www.sri.com/sites/default/files/publications/structure_and_performance_of_dependency_language_model.pdf.

H. Chen, S. Huang, D. Chiang, X. Dai, and J. Chen. Combining character and word information in neural machine translation using a multi-level attention. 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 1284–1293, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1116. URL https://www.aclweb.org/anthology/N18-1116.

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

C. Conforti, M. Huck, and A. Fraser. Neural morphological tagging of lemma sequences for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 39–53. Association for Machine Translation in the Americas, 2018. URL http://aclweb.org/anthology/W18-1805.

M. R. Costa-jussà and J. A. R. Fonollosa. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2058. URL https://www.aclweb.org/anthology/P16-2058.

M. R. Costa-jussà, C. Escolano, and J. A. R. Fonollosa. Byte-based neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 154–158. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/W17-4123.

M. R. Costa-Jussà, N. Casas , C. Escolano, and J. A. R. Fonollosa. Chinese-Catalan: A neural machine translation approach based on pivoting and attention mechanisms. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4):1–8, 2019. URL https://dl.acm.org/doi/abs/10.1145/3312575.

M. R. Costa-jussà, N. Casas , and J. A. Fonollosa. English-Catalan neural machine translation in the biomedical domain through the cascade approach. In *Proceedings of the Multilingual Biomedical Text Processing Workshop of the 11th Language Resources and Evaluation Conference of the European Language Resources Association*, 2018. URL http://temu.bsc.es/multilingualbio2018/wp-content/uploads/2018/03/LREC-2018-PROCEEDINGS-MultilingualBIO.pdf.

Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://www.aclweb.org/anthology/P19-1285.

Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/dauphin17a.html.

A. De Gispert and J. B. Marino. Catalan-English statistical machine translation without parallel corpus: bridging through Spanish. In *Proc. of the Speech and Language technology for minority languages (SALTMIL) workshop*, 2006. URL http://ixa2.si.ehu.es/saltmil/files/procs2006.pdf.

M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. URL https://www.aclweb.org/anthology/W14-3348/.

J. Devlin, M.-W. Chang, K. Lee, and K. 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/.

C. Dyer, A. Kuncoro, M. Ballesteros, and N. 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.

D. Emelianenko, E. Voita, and P. Serdyukov. Sequence modeling with unconstrained generation order. In *Advances in Neural Information Processing Systems 32*, pages 7698–7709. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8986-sequence-modeling-with-unconstrained-generation-order.

C. Escolano, M. R. Costa-jussà, and J. A. R. Fonollosa. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2033. URL https://www.aclweb.org/anthology/P19-2033.

T. Etchegoyhen, A. Azpeitia, and N. Pérez. Exploiting a large strongly comparable corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3523–3529, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1560.

A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://www.aclweb.org/anthology/P18-1082.

O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Yarman Vural, and K. 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, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1026. URL https://www.aclweb.org/anthology/D16-1026.

J. A. R. Fonollosa, N. Casas , and M. R. Costa-jussà. Joint source-target self attention with locality constraints. 2019. URL https://arxiv.org/abs/1905.06596. Under review.

M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144, 2011. URL https://doi.org/10.1007/s10590-011-9090-0.

M. Fortescue, M. Mithun, and N. Evans. *The Oxford Handbook of Polysynthesis*. Oxford Handbooks. Oxford University Press, 2017. ISBN 9780191506192. URL https://books.google.es/books?id=67M1DwAAQBAJ.

W. N. Francis and H. Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979. URL http://icame.uib.no/brown/bcm.html.

Q. Gao and S. Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W08-0509.

Y. Gao, N. I. Nikolov, Y. Hu, and R. H. Hahnloser. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Associ-*

ation for Computational Linguistics, pages 1591–1604, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.145. URL https://www.aclweb.org/anthology/2020.acl-main.145.

M. Garcıa-Martınez, L. Barrault, and F. Bougares. Factored neural machine translation architectures. In *Proceedings of the International Workshop on Spoken Language Translation. Seattle, USA, IWSLT*, volume 16, 2016. URL https://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_2.pdf.

M. García-Martínez, W. Aransa, F. Bougares, and L. Barrault. Addressing data sparsity for neural machine translation between morphologically rich languages. *Machine Translation*, 34(1):1–20, 2020. doi: 10.1007/s10590-019-09242-9. URL https://doi.org/10.1007/s10590-019-09242-9.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017. URL http://proceedings.mlr.press/v70/gehring17a.html.

M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. 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 6114–6123, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1633.

S. Goldwater and D. McClosky. Improving statistical MT through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683, Vancouver, British Columbia, Canada, Oct. 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/H05-1085.

A. Graves. Sequence transduction with recurrent neural networks. In *Representation Learning Workshop, ICML*, 2012. URL https://arxiv.org/abs/1211.3711.

J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=B1l8BtlCb.

J. Gu, H. Hassan, J. Devlin, and V. O. 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 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1032. URL https://www.aclweb.org/anthology/N18-1032.

J. Gu, Q. Liu, and K. Cho. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676, 2019a. doi: 10.1162/tacl_a_00292. URL https://doi.org/10.1162/tacl_a_00292.

J. Gu, C. Wang, and J. Zhao. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32*, pages 11179–11189. Curran Associates, Inc., 2019b. URL http://papers.nips.cc/paper/9297-levenshtein-transformer.

T. He, X. Tan, Y. Xia, D. He, T. Qin, Z. Chen, and T.-Y. Liu. Layer-wise coordination between encoder and decoder for neural machine translation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7955–7965. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8019-layer-wise-coordination-between-encoder-and-decoder-for-neural-machine-translation.pdf.

K. Heafield. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July 2011. URL https://kheafield.com/papers/avenue/kenlm.pdf.

K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013. URL https://kheafield.com/papers/edinburgh/estimate_paper.pdf.

C. D. V. Hoang, R. Haffari, and T. Cohn. Improving neural translation models with linguistic factors. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 7–14, Melbourne, Australia, Dec. 2016. URL https://www.aclweb.org/anthology/U16-1001.

S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91(1), 1991. URL http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

N. Jan, R. Cattoni, S. Sebastian, M. Cettolo, M. Turchi, and M. Federico. The IWSLT 2018 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–6, 2018. URL https://cris.fbk.eu/retrieve/handle/11582/316442/25776/iwslt18-overview.pdf.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. URL http://aclweb.org/anthology/Q17-1024.

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

P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3250.

P. Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

P. Koehn and R. Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://www.aclweb.org/anthology/W17-3204.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://www.aclweb.org/anthology/N03-1017.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

A. Kunchukuttan and P. Bhattacharyya. Orthographic syllable as basic unit for SMT between related languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1912–1917, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1196. URL https://www.aclweb.org/anthology/D16-1196.

A. Kunchukuttan and P. Bhattacharyya. Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4102. URL https://www.aclweb.org/anthology/W17-4102.

B. Lamiroy and R. Gebruers. Syntax and machine translation: The metal project. *Lingvisticae Investigationes*, 13(2):307–332, 1989. URL https://www.jbe-platform.com/content/journals/10.1075/li.13.2.06lam.

G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkYTTf-AZ.

C. Lawrence, B. Kotnis, and M. Niepert. Attending to future tokens for bidirectional sequence generation. In *Proceedings of the 2019 Conference on Empir-*

ical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1–10, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D19-1001.

J. Lee, E. Mansimov, and K. Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1149. URL https://www.aclweb.org/anthology/D18-1149/.

M. P. Lewis, editor. *Ethnologue: Languages of the World*. SIL International, Dallas, TX, USA, sixteenth edition, 2009.

B. Li, A. Drozd, T. Liu, and X. Du. Subword-level composition functions for learning word embeddings. In *Proceedings of the Second Workshop on Subword/Character Level Models*, pages 38–48, New Orleans, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1205. URL https://www.aclweb.org/anthology/W18-1205.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002. URL https://dl.acm.org/citation.cfm?id=1118117.

M.-T. Luong and C. D. Manning. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1100. URL http://www.aclweb.org/anthology/P16-1100.

T. Luong, H. Pham, and C. 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, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL http://www.aclweb.org/anthology/P/P14/P14-5010.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993. ISSN 0891-2017.

S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyyGPP0TZ.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In T. Kobayashi, K. Hirose, and S. Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010. URL http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011. URL https://ieeexplore.ieee.org/document/5947611.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. URL https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

P. Mirowski and A. Vlachos. Dependency recurrent neural language models for sentence completion. 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 511–517, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2084. URL https://www.aclweb.org/anthology/P15-2084/.

J. Nivre, M. Abrams, Ž. Agić, et al. Universal dependencies 2.4, 2019. URL http://hdl.handle.net/11234/1-2988. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

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

P. Passban. *Machine translation of morphologically rich languages using deep neural networks*. PhD thesis, Dublin City University, 2017. URL http://doras.dcu.ie/22200/.

N.-Q. Pham, G. Kruszewski, and G. Boleda. Convolutional neural network language models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1153–1162, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1123. URL https://www.aclweb.org/anthology/D16-1123.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://www.aclweb.org/anthology/W18-6319.

A. Radford. Improving language understanding by generative pre-training. 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

S. Riezler and J. T. Maxwell. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and*

Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 57–64, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W05-0908.

H. Schwenk and M. Douze. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2619. URL https://www.aclweb.org/anthology/W17-2619.

R. Sennrich and B. 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, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2209. URL https://www.aclweb.org/anthology/W16-2209.

R. Sennrich, G. Schneider, M. Volk, and M. Warin. A new hybrid dependency parser for German. *Proceedings of the German Society for Computational Linguistics and Language Technology*, 115:124, 2009. URL https://doi.org/10.5167/uzh-25506.

R. Sennrich, M. Volk, and G. Schneider. Exploiting synergies between open resources for German dependency parsing, POS-tagging, and morphological analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 601–609, 2013. URL https://www.aclweb.org/anthology/R13-1079/.

R. Sennrich, B. Haddow, and A. Birch. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, Aug. 2016a. Association for Computational Linguistics. doi: 10.18653/v1/W16-2323. URL https://www.aclweb.org/anthology/W16-2323.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, Aug. 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016c. Association for Computational Linguistics. doi:

10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

M. A. B. Shaik, A. E.-D. Mousa, R. Schlüter, and H. Ney. Hybrid language models using mixed types of sub-lexical units for open vocabulary german LVCSR. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011. URL https://www.isca-speech.org/archive/archive_papers/interspeech_2011/i11_1441.pdf.

L. Shen, J. Xu, and R. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, 2008. URL https://www.aclweb.org/anthology/P08-1066/.

Y. Shen, Z. Lin, C. wei Huang, and A. Courville. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkgOLb-0W.

Y. Shen, S. Tan, A. Sordoni, and A. Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1l6qiR5F7.

N. Shuyo. Language detection library for java, 2010. URL http://code.google.com/p/language-detection/.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

K. Song, Y. Zhang, M. Zhang, and W. Luo. Improved English to russian translation by neural suffix prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/download/16484/15714.

M. Stern, W. Chan, J. Kiros, and J. Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5976–5985, 2019. URL http://proceedings.mlr.press/v97/stern19a.html.

A. Sundetova, A. Karibayeva, and U. Tukeyev. Structural transfer rules for Kazakh-to-English machine translation in the free/open-source platform Apertium. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 7 (2):48–53, 2014. URL https://dergipark.org.tr/en/download/article-file/395223.

I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011. URL https://www.cs.utoronto.ca/~ilya/pubs/2011/LANG-RNN.pdf.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. URL https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf.

A. Tamchyna, M. Weller-Di Marco, and A. Fraser. Modeling target-side inflection in neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 32–42, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4704. URL https://www.aclweb.org/anthology/W17-4704.

L. Tan, J. Dehdari, and J. van Genabith. An awkward disparity between BLEU / RIBES scores and human judgements in machine translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 74–81, Kyoto, Japan, Oct. 2015. Workshop on Asian Translation. URL https://www.aclweb.org/anthology/W15-5009.

D. Torregrosa, N. Pasricha, M. Masoud, B. R. Chakravarthi, J. Alonso, N. Casas , and M. Arcan. Leveraging rule-based machine translation knowledge for under-resourced neural machine translation models. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 125–133, Dublin, Ireland, Aug. 2019. European Association for Machine Translation. URL https://www.aclweb.org/anthology/W19-6725.

B. Tubay and M. R. Costa-jussà. Neural machine translation with the transformer and multi-source romance languages for the biomedical WMT 2018 task. In *Proceedings of the Third Conference on Machine Translation: Shared*

Task Papers, pages 667–670, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6449.

N. Casas , C. Escolano, M. R. Costa-jussà, and J. A. R. Fonollosa. The TALP-UPC machine translation systems for WMT18 news shared translation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 355–360, Belgium, Brussels, Oct. 2018a. Association for Computational Linguistics. doi: 10.18653/v1/W18-6406. URL https://www.aclweb.org/anthology/W18-6406.

N. Casas , J. A. Fonollosa, and M. R. Costa-jussà. A differentiable BLEU loss. Analysis and first results. Presented at the Workshop of the International Conference on Learning Representations (ICLR), 2018b. URL https://openreview.net/forum?id=HkG7hzyvf.

N. Casas , J. A. R. Fonollosa, C. Escolano, C. Basta, and M. R. Costa-jussà. The TALP-UPC machine translation systems for WMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5311. URL https://www.aclweb.org/anthology/W19-5311.

N. Casas , M. R. Costa-jussà, and J. A. R. Fonollosa. Sparsely factored neural machine translation. 2020a. Under review.

N. Casas , M. R. Costa-jussà, and J. A. R. Fonollosa. Combining subword representations into word-level representations in the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 66–71, Online, July 2020b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-srw.10.

N. Casas , M. R. Costa-jussà, J. A. R. Fonollosa, J. A. Alonso, and R. Fanlo. Linguistic knowledge-based vocabularies for neural machine translation. *Natural Language Engineering*, page 1–22, 2020c. URL https://doi.org/10.1017/S1351324920000364.

N. Casas , J. A. R. Fonollosa, and M. R. Costa-jussà. Syntax-driven iterative expansion language models for controllable text generation. 2020d. Accepted for publication at the EMNLP 2020 Workshop on Structured Prediction for NLP.

M. Utiyama and H. Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, 2007. URL https://www.aclweb.org/anthology/N07-1061/.

C. Vania and A. Lopez. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2016–2027. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1184. URL http://aclweb.org/anthology/P17-1184.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

S. Virpioja, P. Smit, S.-A. Grönroos, M. Kurimo, et al. Morfessor 2.0: Python implementation and extensions for morfessor baseline. 2013. URL http://urn.fi/URN:ISBN:978-952-60-5501-5.

X. Wang, H. Pham, P. Arthur, and G. Neubig. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Skeke3C5Fm.

S. Welleck, K. Brantley, H. D. III, and K. Cho. Non-monotonic sequential text generation. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6716–6726, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/welleck19a.html.

R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. URL https://doi.org/10.1162/neco.1989.1.2.270.

F. Wu, A. Fan, A. Baevski, Y. Dauphin, and M. Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkVhlh09tX.

H. Wu and H. Wang. Pivot language approach for phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the As-*

sociation of Computational Linguistics, pages 856–863, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-1108.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. URL https://arxiv.org/abs/1609.08144.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5754–5764. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.

W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014. URL https://arxiv.org/abs/1409.2329.

J. Zhao, S. Mudgal, and Y. Liang. Generalizing word embeddings using bag of subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 601–606, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1059.

Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100. ACM, 2018. URL https://doi.org/10.1145/3209978.3210080.

M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1561.

B. Zoph and K. Knight. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1004. URL https://www.aclweb.org/anthology/N16-1004.

A. International MT Evaluation Campaigns

One of the main focuses of this thesis is MT. This way, the participation in international MT evaluation campaigns can be seen as the practical complement of the pure research aspect.

In this chapter, we describe the participation of our research group, TALP, from the Polytechnic University of Catalonia (UPC), in the evaluation campaigns of the third and fourth conferences on machine translation, previously known as Workshop on Machine Translation (WMT). In both participations, the preparation of the submitted translation system and the article writing was led by the author of this thesis.

Note that the techniques proposed in previous chapters to imbue linguistic knowledge in neural systems were not applied in these WMT participations. The cause is twofold. First, these techniques require external resources to extract the linguistic knowledge, which would make the submissions non-constrained, excluding them from the main competition. Second, we understood that there conditions for the proposed techniques to provide an improvement in the final translation quality were not met. Therefore, we decide not to make use of them and instead use the most appropriate tools for the situation.

A.1. Third Conference on Machine Translation (WMT18)

The Third Conference on Machine Translation (WMT18)[1] was co-located with EMNLP18. Our participation, which is described in (Casas et al., 2018a), focused on the news translation shared task, specifically in the multilingual subtrack, translating Finnish and Estonian to and from English. Both can be considered low-resource languages in general, and also in particular for this shared task, based on the volume of data made available for training, especially Estonian.

[1] http://www.statmt.org/wmt18/

Finnish and Estonian are respectively the official languages of Finland and Estonia, having 5.4 and 1.1 million native speakers (Lewis, 2009). They are **Finnic Languages**, a branch within the Uralic Language family.

Estonian and Finnish make use of the Latin alphabet with some additional letters, each one incorporating extra letters (e.g. ä, ö, ü, õ, š, ž).

Finnish and Estonian are morphologically-rich **agglutinative languages**. Estonian presents fourteen grammatical cases while Finnish presents fifteen. Verb conjungations are very regular in both languages. Neither of them has grammatical gender nor definite or indefinite articles. Both have flexible word order, but the basic order is subject-verb-object.

Like other Finnic languages, both Finnish and Estonian present consonant gradation (consonants are classified in grades according to phonologic criteria, and such grades condition the combined appearance of the consonants in a derived word), but the gradation patterns each one follows are different.

While Finnish has kept most of its late Proto-Finnic linguistic traits, Estonian has lost some of its former characteristics, like vowel harmony (vowels in a word cannot appear freely but their allowance is constrained by rules), which in Finnish affects case and derivational endings. Also, Estonian mostly lost the word-final sound, making its inflectional morphology more fusional for nouns and adjectives (Fortescue et al., 2017). German language influence also led Estonian to use more postpositions where Finnish uses cases. Geographical location has also led to differences in the loanwords borrowed by each language.

A.1.1. Motivation

The application of NMT to low resource language pairs needs extra techniques to achieve good translation quality. These are some of the frequently used approaches:

Back-translation (Sennrich et al., 2016b) consists in training an auxiliary translation system from target language to source language and use it to translate a large target language monolingual corpus into theç source language, and then use such synthetic source-target sentence pairs to augment the originally available parallel corpus and train a new source language to target language translation system on it. Back-translation can be applied iteratively until no further improvement is gained (Artetxe et al., 2020).

Pivoting approaches use a third resource-rich language as *pivot* and train translation systems from source language to pivot and from pivot to target language. These auxiliary systems can either be used in *cascade* to obtain source-to-target translations, or be used to build syntethic parallel source-target corpora (i.e. *pseudocorpus approach*). A recent application of pivoting techniques to NMT can be found in (Costa-jussà et al., 2018).

Adversarial learning (Lample et al., 2018; Artetxe et al., 2018) in a multi-task learning setup so that there is an auxiliary text (denoising) auto-encoding task whose internal sentence representation is aligned with the ones from the translation task by means of a discriminator in feature space.

Pre-trained cross-lingual embeddings (Artetxe et al., 2016, 2017) can be used complementarily to further reduce the need for parallel data.

Finding parallel data from a similar source language and the same target language (or vice versa) and adding it to the original parallel corpus. With such a composite training data set, a wordpiece-level vocabulary can leverage the common word stems between the similar languages and profit from the combined amount of data. This approach was used in this submission, as described in Sections A.1.2 and A.1.3.

Multilingual translation. Among the different types of multilingual systems there are the many-to-one approaches and the many-to-many approaches. The former is aiming to translate to one single language and can simply concatenate source languages (Zoph and Knight, 2016; Tubay and Costa-jussà, 2018). However, the latter either needs to use independent encoders and decoders (Schwenk and Douze, 2017; Firat et al., 2016; Escolano et al., 2019) or when using universal encoder and decoders (Johnson et al., 2017) needs to add a tag in the source input to let the system know to which language it is translating. This many-to-many systems are an alternative to pivot systems. However, most these multilingual systems are not able to achieve the level of performance of pivot systems yet.

A.1.2. Corpora and Data preparation

All proposed systems in our WMT18 participation were constrained, using exclusively parallel data provided by the organization. For the English - Finnish language pair the data employed was the Europarl corpus version 7 and 8, Paracrawl corpus, Rapid corpus of EU press releases and Wiki Headlines corpus. For the English - Estonian data the Europarl v8 corpus, Paracrawl and Rapid corpus of EU press releases corpus were employed.

All language pairs have been preprocessed following the proposed scripts by the organization of the conference. The pipeline consisted in normalizing punctuation, tokenization and truecasing using the standard Moses (Koehn et al., 2007) scripts. With the addition that, for tokenization, no escaping of special characters was performed.

For the language pair of English - Estonian we found that from Paracrawl corpus a considerable number of sentences were not suitable sentences in the intended languages, but apparently random sequences of upper case characters. In order to remove them, an additional step of language detection was performed using library `langdetect`[2], which is a port to Python of library `language-detection` (Shuyo, 2010). The criteria for removing noisy sentences from the dataset was that either one of the languages of the pair could not be identified as a language.

The sizes of the different data sets compiled for each language pair and once cleaned as described earlier in this section are presented in Table A.1.

corpus	lang	set	sentences	words
En-Et	*En*	train	998547	23056922
		test	2000	44305
	Et	train	998547	17376004
		test	2000	34733
En-Fi	*En*	train	3064124	62208347
		dev	3000	64611
		test	3002	63417
	Fi	train	3064124	45692989
		dev	3000	48839
		test	3002	46572

Table A.1.: Corpus statistics in number of sentences and words for both parallel corpora, English - Estonian and English - Finnish.

As Finnish and Estonian belong to the Finnic language family and are similar to each other, we aimed at combining the individual parallel corpora (*En - Fi* and *En - Es*) into a single larger corpus. For the translation directions where English is the target language (i.e. $Fi \rightarrow En$ and $Et \rightarrow En$) we prepared a combined $Fi + Et \rightarrow En$ corpus by simply concatenating the original ones. This approach was not applicable to the reverse directions, as we needed some way to convey the information about whether to generate either Finnish or Estonian as part of the input to the neural network. Following the approach in (Johnson et al., 2017), we modified the individual parallel corpora to add a prefix to the English sentences to mark whether the associated target sentence was Finnish or Estonian, and

[2] https://github.com/Mimino666/langdetect

then proceed to concatenate both corpora into the final combined one *En* → *Fi* + *Et*. The prefixes used were respectively `<fi>` and `<et>`. This prefix needs to be added likewise to the test English sentences when decoding them into Finnish or Estonian.

As the combined corpora are concatenations of the individual ones, their sizes can be computed from the figures in Table A.1 by mere addition of the individual sizes of each language pair.

A.1.3. System Description

In this section we present the translation systems used for our participation, both in terms of vocabulary extraction strategies followed (Section A.1.3), of neural architecture used (Section A.1.3) and of needed post-processing (Section A.1.3).

Vocabulary Extraction

The NMT models used for our submissions to the shared task, which are described in Section A.1.3 made use of pre-defined sets of discrete tokens that comprise the *vocabulary*.

The vocabulary of each of our translation systems (both the final submissions and the systems trained for reference described in Section A.1.4) was based on word-piece extraction (Wu et al., 2016). For each system, the source and target vocabularies were extracted separately, aiming at a vocabulary size of 32K tokens. Vocabularies are not shared between source and target languages in any case.

Word-piece vocabularies (or the very similar Byte-Pair Encoding (BPE) vocabularies (Sennrich et al., 2016c)) are usually applied to extract vocabularies from corpora that contain data from similar languages in order to try to find common stems and derivational suffixes so that the language commonalities can be leveraged by the neural network training.

NMT Models

All the submissions presented to the shared task made use of the Transformer NMT architecture, which is described in Section 2.3.2. We used the implementation released by the authors of (Vaswani et al., 2017) [3]

[3]The authors of (Vaswani et al., 2017) made the source code available at `https://github.com/tensorflow/tensor2tensor`. For this participation, version `1.2.9` was used.

The complete hyperparameter configuration used for all the attention-based neural machine translation models in our submissions (which consisted in the `transformer_base` parameter set in `tensor2tensor`) is shown in Table A.2.

hyperparameter	**value**
attention layers	6
attention heads per layer	8
hidden size (embedding)	512
batch size (in tokens)	4096 (4 GPU)
training steps	800000
tokenization strategy	wordpiece
vocabulary size	32K
optimization algorithm	Adam
learning rate	warmup + decay

Table A.2.: Hyperparameters of the neural model.

After the training, the weights of the last 5 checkpoints (having checkpoints stored every 2000 optimization steps) are averaged to obtain the final model.

Post-processing

Following the inverse steps of the processing described in Section A.1.2, the decoded outputs of NMT model need to be de-truecased and de-tokenized by means of the appropriate *Moses* scripts.

A.1.4. Experiments

The hypothesis on which we based this work was that, given the similarity between Estonian and Finnish, a system trained with the combination of the data from both languages would outperform systems trained on the individual language datasets.

In order to validate this hypothesis, we conducted direct experiments, training systems on the individual language datasets and also on the combined datasets (as described in Section A.1.2), and comparing their translation quality. The datasets used for testing the performance were `newsdev2018` for Estonian - English and `newstest2017` for Finnish - English. The results of the experiments are shown in Table A.3, were all figures represent case-insensitive BLEU score over the aforementioned reference test corpora.

direction	individual	combined	ΔBLEU
En → *Fi*	24.36	25.21	+0.85
Fi → *En*	29.39	30.00	+0.61
En → *Et*	15.97	18.92	+2.95
Et → *En*	21.66	25.66	+4.00

Table A.3.: Comparison between translation quality (case-insensitive BLEU) of systems trained on the individual language data vs. systems trained on the combined data.

While the results for Finnish are not very different between the individual and combined data trainings [4], the results for Estonian show an important improvement of the training on the combined data over the individual data. This correlates with the fact that the Estonian - English training set is less than one third the size of the Finnish - English, therefore the size increase in the Finnish - English combined training corpus is much smaller than the increase for Estonian - English, as shown in Table A.1.

A.1.5. Conclusions

Our experiments in the WMT18 participation, suggested that for low resource languages, enlarging the training data with translations from a similar language can lead to important improvements in the translation quality when using subword-level vocabulary extraction strategies.

English→ Finnish

	Ave. %	Ave. z	System
1	64.7	0.521	NICT
	63.1	0.466	HY-NMT
3	59.2	0.324	UEDIN
	58.3	0.271	AALTO
	57.9	0.258	HY-NMT-2STEP
	57.4	0.238	TALP-UPC
	55.9	0.184	CUNI-KOCMI
	56.6	0.183	ONLINE-B
9	45.9	−0.212	ONLINE-A
	45.3	−0.233	ONLINE-G
11	42.7	−0.334	HY-SMT
	41.5	−0.369	HY-AH

Finnish→ English

	Ave. %	Ave. z	System
1	75.2	0.153	NICT
	74.4	0.128	HY-NMT
	74.0	0.103	UEDIN
	72.7	0.083	CUNI-KOCMI
	72.9	0.078	ONLINE-B
	71.9	0.047	TALP-UPC
	71.5	0.045	ONLINE-A
8	66.1	−0.134	ONLINE-G
9	58.9	−0.404	JUCBNMT

Figure A.1.: Results of the TALP participation in WMT18 News Translation Shared Task for Finnish ⟷ English.

The results of the TALP participation in the WMT18 News Translation Shared Task are shown in Figures A.1 and A.2. Our submission for Finnish→English was

[4] Improvements of less than 1 BLEU point are normally considered neglectable.

English→ Estonian

	Ave. %	Ave. z	System
1	64.9	0.549	TILDE-NC-NMT
2	62.1	0.453	NICT
	61.6	0.427	TILDE-C-NMT
	61.2	0.418	TILDE-C-NMT-2BT
5	58.6	0.340	AALTO
	58.6	0.329	HY-NMT
	57.5	0.295	UEDIN
8	55.5	0.216	CUNI-KOCMI
	54.6	0.181	TALP-UPC
10	52.1	0.097	ONLINE-B
11	45.7	−0.132	NEUROTOLGE.EE
12	43.8	−0.195	ONLINE-A
13	37.6	−0.406	ONLINE-G
14	34.3	−0.520	PARFDA

Estonian→ English

	Ave. %	Ave. z	System
1	73.3	0.326	TILDE-NC-NMT
2	71.1	0.238	NICT
	69.9	0.215	TILDE-C-NMT
	69.0	0.187	TILDE-C-NMT-2BT
	69.2	0.186	UEDIN
	68.7	0.171	TILDE-C-NMT-COMB
	67.1	0.117	ONLINE-B
	66.4	0.106	HY-NMT
	66.8	0.106	TALP-UPC
10	65.4	0.063	ONLINE-A
	64.0	0.007	CUNI-KOCMI
12	59.4	−0.117	NEUROTOLGE.EE
13	52.7	−0.341	ONLINE-G
14	34.6	−0.950	UNSUPTARTU

Figure A.2.: Results of the TALP participation in WMT18 News Translation Shared Task for Estonian ⟷ English.

featured in the first cluster of systems, while our submissions for English→Finnish and Estonian→English were featured in the second cluster of systems, meaning that their translations were not statistically distinguishable from any of the submissions in the same cluster.

A.2. Fourth Conference on Machine Translation (WMT19)

The Third Conference on Machine Translation (WMT18)[5] was co-located with EMNLP18. Our participation, which was described extensively in (Casas et al., 2019), focused on the news translation shared task, specifically in the low resource language pair Kazakh - English. The amount of available parallel Kazakh-English data was very low. In order to overcome this problem in the frame of the shared task, we made use of Russian as an pivot language. This way, we used English-Russian and Kazakh-Russian data to train intermediate translation systems that we then used to create synthetic pseudo-parallel Kazakh-English data. This data enabled us to train the final Kazakh-English translation systems.

There are many techniques that can be applied to a low resource NMT scenario. The most relevant ones are described in Section A.1.1. In the frame of the WMT19 news translation shared task several of them were applicable:

An English+Russian→Kakakh multilingual system could be trained, but the amount of Kazakh-Russian data is much larger than Kazakh-English, which would bias the encoder toward Russian; as Russian is not similar to English this

[5] http://www.statmt.org/wmt19/

would decrease the effectiveness of the approach, as opposed to what happens for similar languages (Casas et al., 2018a).

Back-translation could also be applied in this context, but the amount of Kazakh monolingual data is not very large and it is crawled data, with presumably low quality. It could have been used additionally to other techniques, though.

Finally, pivoting approaches are also applicable to this scenario. The cascade approach, however, would not allow to profit from the existing parallel English-Kazakh data, making the pseudo-parallel corpus approach the most sensible option.

A.2.1. Corpora and Data Preparation

In order to train our MT systems, we used the data made available by the shared task organizers, including the not only Kazakh-English data but also the English-Russian and Kazakh-Russian data to train pivot translation systems. In this section we describe the data used for each language pair and the processing applied to each of them in order to compile appropriate training datasets.

Kazakh-English

The available parallel Kazakh-English corpora for the shared task included News Commentary v14, Wiki Titles v1 and a crawled corpus prepared by Bagdat Myrzakhmetov of Nazarbayev University.

Wiki Titles accounted for half of the available parallel segments, but its sentences were around 2 tokens long in average. Therefore, we decided not to include it in the training data, to avoid biasing the trained systems toward short translations.

After concatenating the training corpora, we used the standard Moses scripts to preprocess them, including tokenization, truecasing and cleaning. The statistics of the resulting training data are shown in table A.4.

Lang.	Sents.	Words	Vocab.	L_{max}	L_{mean}
Kazakh	99.6K	1.2M	139.6K	85	11.7
English		1.5M	85.3K	102	14.9

Table A.4.: Summary statistics of the Kazakh-English training data.

The WMT organization split a part of News Commentary to use as development[6].

[6]The part of News Commentary provided as development data was excluded from the training set.

From this data, we left 500 parallel sentences as hold-out to assess final system translation quality and left the remaining 1566 segments as development data.

English-Russian

The available parallel English-Russian corpora for the shared task included News Commentary v14, Wiki Titles v1, Common Crawl corpus, ParaCrawl v3, Yandex Corpus and the United Nations Parallel Corpus v1.0 (Ziemski et al., 2016).

Following the rationale exposed for the English-Kazakh Wiki Titles data, we also dropped the English-Russian Wiki Titles data.

Among the other corpora, some are of very large size. In order to assemble a manageable final training dataset and taking into account the high presence of garbage in the crawled datasets, before combining the individual corpora, we filtered each corpus and selected from each a random sample of segments.

For the filtering, we applied heuristic criteria based on our visual inspection of the data, including elimination of lines with repeated separation characters (like `++++` or `----`), elimination of fixed expressions (like "The time is now", which appeared several times in some corpora) and eliminating lines with high ratio of numbers and punctuation characters.

For the random sample, from UN Corpus we took 2M segments out of 23M, from Common Crawl we took 200K out of 900K, from ParaCrawl we took 4M out of 12M and from the Yandex Corpus we took all the 1M segments. These samples were then combined and went through standard processing with Moses scripts, including tokenization, truecasing and cleaning. After combining them, we applied Moses corpus cleaning with more aggressive settings (sentences between 5 and 80 words and a maximum length ratio of 3.0 between source and target). From the combined corpus, we extracted 4000 random lines as development data and 1000 segments as hold out test set, leaving the rest for training. The statistics of the resulting training data are shown in table A.5.

Lang.	Sents.	Words	Vocab.	L_{max}	L_{mean}
Russian	6.1M	125.6M	3.2M	80	20.7
English		144.9M	2.0M	80	23.9

Table A.5.: Summary statistics of the English-Russian training data.

Kazakh-Russian

The available parallel Kazakh-Russian corpora for the shared task included News Commentary v14 and a crawled Russian-Kazakh corpus prepared by Bagdat Myrzakhmetov of Nazarbayev University.

After concatenating the training corpora, we used the Moses scripts for preprocessing, including tokenization, truecasing and cleaning, using the same settings as for the aggressive English-Russian data cleaning described before. From the combined corpus, we extracted 4000 lines as development data and 1000 segments as hold out test set, leaving the rest for training. The statistics of the resulting training corpus are shown in table A.6.

Lang.	Sents.	Words	Vocab.	L_{max}	L_{mean}
Russian	4.2M	78.8M	1.4M	96	18.9
Kazakh		75.3M	1.6M	70	18.0

Table A.6.: Summary statistics of the Russian-Kazakh training data.

A.2.2. System Description

The amount of available parallel training data for English-Kazakh was scarce. When an NMT system was directly trained on this data, the resulting translation quality was very low, as shown in Section A.2.3.

Given the amount of available English-Russian and Kazakh-Russian parallel training data, we decided to use Russian as pivot language. Taking into account the availability of some parallel Kazakh-English data, the pivoting approach that best suits this case is to prepare pseudo-parallel English-Kazakh and Kazakh-English corpora based on the Russian data and then combine it with the parallel English-Kazakh data. Further justification of the technique used can be found at the end of Section A.2.

In pivoting approaches, the final translation quality does not get influenced significantly if synthetic data is used for the source language side; on the other hand, using synthetic data for the target language side results in degraded translation quality in the final system (Costa-jussà et al., 2018; Costa-Jussà et al., 2019). Therefore, we will create two different pseudo-parallel corpora for English→Kazakh and Kazakh→English.

In order to create the English→Kazakh synthetic data, we translated the Russian side of the Russian-Kazakh corpus into English. To perform this translation, we

need an intermediate Russian→English system. We made use of the Russian-English corpus to train this pivot system.

In order to create the Kazakh→English synthetic data, we translated the Russian side of the Russian-English corpus into Kazakh. To perform this translation, we need an intermediate Russian→Kazakh system. We made use of the Russian-Kazakh corpus to train this pivot system.

The preparation and training of the two pivot translation systems is further described in Section A.2.2

Once the synthetic data was prepared by means of the pivot translation systems, we combined each synthetic corpus with the parallel data, obtaining the respective training datasets for the two translation directions. This is further described in Section A.2.2.

Finally, we trained the English→Kazakh and Kazakh→English translation systems on the previously described mix of parallel and synthetic corpora. The NMT model used is presented in Section A.2.2.

Pivot SMT Systems

For the Russian→English and Russian→Kazakh pivot translation systems we decided to use Moses (Koehn et al., 2007). The use of pivot approaches for SMT has been studied previously, like the works by De Gispert and Marino (2006), Wu and Wang (2007) or Utiyama and Isahara (2007). Another option would have been to use a Neural Machine Translation (NMT) approach, but this would have required large amounts of GPU time to translate the pseudo-parallel corpora.

While the English language presents simple morphology, Russian is morphologically rich and Kazakh is agglutinative. Therefore, the amount of surface forms in a word-level vocabulary of the two latter languages is very high. This way, we decided to apply subword-level tokenization before training the SMT systems. For this, we used Byte-Pair Encoding (BPE) (Sennrich et al., 2016c) to extract a vocabulary of subword parts based on frequency statistics. We prepared separate BPE vocabularies for each language, with 32K merge operations each. Although not frequent, there are some precedents for subword tokenization in SMT, like the work by Kunchukuttan and Bhattacharyya (2016, 2017).

The use of subword tokenization leads to longer token sequence lengths compared to the usual word-based vocabularies of SMT systems. In order to cope with this fact, we configured the subword-based SMT systems to have longer n-gram order for their Language Models (LM) and phrase tables: the typical n-gram order used is 3 and we used 6. All other Moses configuration settings are the

standard ones, using KenLM as language model (Heafield, 2011; Heafield et al., 2013) and MGIZA++ (Gao and Vogel, 2008) for alignment.

The data used to create the respective target-side LMs consisted of the target side of the parallel data used for training. Some improvement could have been gained by using the available extra monolingual English and Kazakh data for the LMs.

Combination of Parallel and Synthetic Data

The process followed to combine the parallel data with the synthetic data was the same for English-Kazakh and for Kazakh-English: we oversampled at 300% the parallel data and concatenated it with the synthetic data, obtaining the final training datasets on which the translation systems for the submissions were trained.

Joint Source-Target Self-Attention NMT

The translation system trained on the augmented Kazakh-English data and used for the final WMT submissions is based on the architecture proposed by (He et al., 2018; Fonollosa et al., 2019). This approach is based on the self-attention blocks from (Vaswani et al., 2017), but breaks from the encoder-decoder structure and has only a single decoder block that is fed both the source and target sentences, therefore learning joint source-target representations from the initial layers. This model resembles how a language modeling architecture is trained and used for inference.

The positional encodings are applied separately to source and target. An extra embedded vector representation is added to the combination of token and position in order to distinguish source and target parts.

The attention weights can be masked to control the receptive fields (Fonollosa et al., 2019). Both source-source and target-target receptive fields are constrained to a local window around each token, while target-source receptive fields are unconstrained.

The hyperparameter configuration used was the same as the one originally used by the authors for WMT'14 English-German (14 layers, 1024 as embedding dimensionality, feedforward expansion of dimensionality 4096 and 16 attention heads).

Direction	RBMT	SMT (w)	SMT (sw)	NMT	NMT pseud.
Kazakh→English	1.51	6.34	7.48	2.32	21.00
English→Kazakh	1.46	3.53	3.82	1.42	15.47

Table A.7.: BLEU scores (cased) of the Rule-based baseline (**RBMT**), the Moses system trained on the parallel Kazakh-English data with word-level tokenization (**SMT(w)**), the Moses system trained on the parallel Kazakh-English data with subword-level tokenization (**SMT(sw)**), the **NMT** system trained on the parallel Kazakh-English data, and the final systems trained on the augmented pseudo-parallel corpus data (**NMT pseud.**)

For Kazakh-English we used separate BPE vocabularies with 32K merge operations, while for English-Kazakh we used a joint BPE vocabulary with 32K merge operations, together with shared source-target embeddings.

A.2.3. Experiments and Results

In order to assess the translation quality of the systems, we computed the BLEU score (Papineni et al., 2002) over the respective held out test sets.

As there is not much literature of current NMT approaches being applied to English-Kazakh, we prepared different baselines to gauge the range of BLEU values to expect:

- Rule-based machine translation system (RBMT): we used the Apertium system (Forcada et al., 2011; Sundetova et al., 2014; Assem and Aida, 2013), which is based on transfer rules distilled from linguistic knowledge. Using the BLEU score to compare an RBMT system with data-driven systems is not fair (see (Koehn, 2010) §8.2.7) but we included it to have a broader picture.

- Statistical Machine Translation with word-level tokenization (SMT(w)): we trained a Moses system on the parallel Kazakh-English data, using normal word-level tokenization

- Statistical Machine Translation with subword-level tokenization (SMT(sw)): we trained a Moses system on the parallel Kazakh-English data, using BPE tokenization with 10K merge operations[7]. Moses default values were used for the rest of configuration settings .

[7]The low number of BPE merge operations is justified with the low amount of training data

- Neural Machine Translation (NMT): we trained a Transformer model on the parallel Kazakh-English data, using BPE tokenization with 10K merge operations, separately for source and target. We used the fairseq (Ott et al., 2019) implementation with the same hyperparameters as the IWSLT model, namely an embedding dimensionality of 512, 6 layers of attention, 4 attention heads and 1024 for the feedwordward expansion dimensionality.

The translation quality BLEU scores of the aforedescribed baselines were very low, as shown in table A.7.

In order to evaluate the pivot translation systems described in Section A.2.2, we also measured the BLEU scores in the respective held out test sets, obtaining 36.05 BLEU for the Russian→English system and 21.06 for the Russian→Kazakh system. With these pivot systems, we created two pseudo-parallel synthetic corpora, merged them with the parallel data and trained a self-attention NMT model that obtained BLEU scores one order of magnitude above the chosen baselines, as shown in table A.7.

When we tested the final Kazakh→English system on the shared task test set, we identified several sentences that remained completely in Cyrillic script. In order to mitigate this problem, we trained a SMT system on the augmented Kazakh-English data and used it for the sentences that had a large percentage of Cyrillic characters. This lead to a mere 0.1 increase in the case-insensitive BLEU score and no change for the uncased one.

A.2.4. Conclusion

Our experiments showcased the effectiveness of pivoting approaches for low resourced scenarios, making use of SMT to support the data augmentation process, while using the more effective attention-based NMT approaches for the final translation systems.

The results of the TALP participation in the WMT19 News Translation Shared Task are shown in Figure A.3. Our submission for Kazakh→English was featured in the second cluster of systems, meaning that its translations were not statistically distinguishable from any of the submissions in the same cluster.

Kazakh→ English

Ave.	Ave. z	System
72.2	0.270	online-B
70.1	0.218	NEU
69.7	0.189	rug-morfessor
68.1	0.133	online-G
67.1	0.113	talp-upc-2019
67.0	0.092	NRC-CNRC
65.8	0.066	Frank-s-MT
65.6	0.064	NICT
64.5	0.003	CUNI-T2T-transfer
48.9	−0.477	UMD
32.1	−1.058	DBMS-KU

English→ Kazakh

Ave.	Ave. z	System
81.5	0.746	HUMAN
67.6	0.262	UAlacant-NMT
63.8	0.243	online-B
63.8	0.222	UAlacant-NM
63.8	0.222	RBMT
63.3	0.126	NEU
63.3	0.108	MSRA-CrossBERT
60.4	0.097	CUNI-T2T-transfer
61.7	0.078	online-G
55.2	−0.049	rug-bpe
49.0	−0.328	talp-upc-2019
41.4	−0.493	NICT
11.6	−1.395	DBMS-KU

Figure A.3.: Results of the TALP participation in WMT19 News Translation Shared Task.