UNIVERSITEIT VAN AMSTERDAM

MSc Artificial Intelligence
Master Thesis

Inductive Biases for Morphologically Informed Neural Machine Translation

by

Ivo Verhoeven

13013319

August 3, 2022

48 ECTS
November 2021 - July 2022

Supervisor:
Dr Wilker Aziz

Assessor:
Dr Ekaterina Shutova

Institute for Logic, Language & Computation

Acknowledgements

I am privileged to have had many excellent teachers and mentors. I would like to thank my supervisor, Wilker Aziz. His insights have taught me volumes, and our conversations have been a critical source of motivation. I would also like to thank my examiner, Katia Shutova. I look forward to future collaboration.

Last, I thank my family and friends, to whom I am indebted many kindnesses. I thank Laney for being besides me, always, and Mantou for being as good a late night companion as I could ask for.

Abstract

The words which comprise human language, are themselves complex units. Each carries meaning in isolation, but their structure is often altered to accommodate larger compositions, according to some set of grammatical rules. The processes that determine how the word is formed, and for which occasions it applies, are heavily patterned. Human beings, whether conscious of it or not, can leverage these patterns to quickly generate new word-forms when situations necessitate them. Neural language models, as used in neural machine translation systems, likely do not learn these patterns and show limited capacity in decomposing words as humans would. Instead, they merely learn to associate certain string segments with others, imitating the data used to train them. Morphologically rich languages, with very complex word-formation processes, have word-forms that occur only in rare situations. As a result, translation performance suffers when neural language models are required to produce word forms it has not seen before.

This thesis explores these morphological word formation processes, and how neural language models interact with them. Ultimately, it seeks to adapt pre-trained neural machine translation models, towards greater understanding of morphology. The requirement of pre-trained systems, a practical necessity for many researchers, invalidates previous techniques presented for this task. As such, a novel gradient-based meta-learning framework is formulated, which only alters the sampling method to incorporate morphological information implicitly. This process is coined 'morphological cross-transfer', and separates meaning from function in the learning phase. For this, strong automated morphological analyzers are required. This is covered in detail, and neural systems for this task are re-implemented, before verifying their usage on natural language corpora. A second required component is a measurable notion of morphological competence. This too is covered in some detail, presenting a novel methodology that extends easily to designing task samplers typically found in meta-learning setups. Finally, experiments with morphological cross-transfer indicate slightly improved translation systems, and slightly improved dedicated morphological inflectors, although the objectives are not achieved simultaneously. This opens up new avenues of research into post-hoc adaptation techniques for providing neural language models with desired inductive biases.

Contents

List of Figures

List of Tables

1 Introduction

Neural machine translation (NMT) considers the application deep learning methods and models to translate text in one language, the source, to another, the target. It is an instance of sequence-to-sequence learning (seq2seq), typically consisting of a single model, itself containing a source-language specific encoder, and a target-language specific decoder. The encoder considers the entire source-side string, and compresses it into a variable length hidden representation. The decoder is asked to predict, from the encoder's message (typically pooled by an attention mechanism), and all previously predicted tokens, the next most likely token. Omitting some details, this process is repeated until some boundary is reached, at which point the translation is considered complete. While general, this setup is by no means the only available for generating translations, but since its popularization by Bahdanau, Cho, and Bengio [1], it has seen tremendous success, even with newer, vastly larger models [2].

[1]: Bahdanau et al. (2014), 'Neural machine translation by jointly learning to align and translate'

The properties of the source- and target-side languages need not match. The mismatch of certain properties likely has more effect on generated translations than others. One likely determinant of translation quality is the morphologically complexity of the considered languages. Morphology controls how parts of a word combine to create new words with specific meanings. Morphologically rich or complex languages can attach many new meanings to root words, whereas morphologically poor languages do so through many, less complex words.

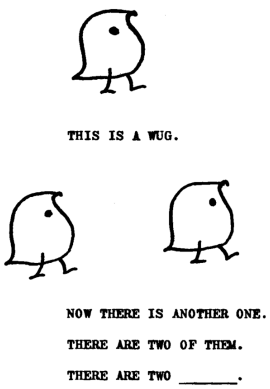


Figure 1.1: Jean Berko's famous 'Wug Test' requires children to infer the plural form of a word with which they have no experience. We ask NMT systems to do the same, but typically in vastly more complex scenarios. Taken from [3].

Unfortunately, historically NMT research is driven by languages either close in morphological complexity, or ones similarly impoverished. An especially prominent culprit is English, a morphologically poor language that dominates parallel corpora. As a result, modern architectures contain no explicit inductive bias towards the word formation processes prevalent in morphologically rich languages. Instead, these systems only associate certain sequence segments with others, with no assumptions

Deep learning systems 'learn' by applying patterns found in past experiences. Without past experiences, an inductive bias can extend those patterns to generalize well to novel situations. Specifically for morphologically rich languages, this is crucial, as increased complexity implies new words are likely in new situations. Rephrased, there exists a data sparsity problem. How can a system learn to produce a certain word when the specific form of that word is not present in the training data?

One way to express an inductive bias is by changing the modelling architecture. For image recognition, one dispenses linear layers for shift-invariant `CNN`s [4, 5], and for text classification, one opts for recurrent architectures instead [6]. An inductive bias for morphological inflection, however, is trickier to express algorithmically. Ataman, Aziz, and Birch [7] do so by learning a hierarchical latent-variable model. Beyond the additional computational cost relative to standard NMT architectures, however, this route of reasoning also invalidates pre-trained architectures entirely. With the unprecedented scale of modern deep learning architectures, built with more data, more compute power and more parameters than any single researcher is likely to possess, repeating training becomes

[7]: Ataman et al. (2019), 'A latent morphology model for open-vocabulary neural machine translation'

practically intractable. Ideally, research into morphological awareness starts from a competent NMT system.

Instead, this thesis proposes a post-training fine-tuning procedure. It can take any NMT system, through adaptation it improves its capacity to morphologically inflect. Despite the lack of an explicit inductive bias, hopefully, an implicit one can be learned to allow for more robust generalization to new words.

1.1 Outline & Contributions

2. **Morphological Tagging and Lemmatization in Context**: this chapter starts with an overview of morphology. Some crucial definitions are given and how these are tackled from a modelling perspective is outlined. The second section covers the tasks of jointly lemmatizing and morphologically annotating natural language. This was needed as a pre-processing step, and given no solutions had been made available in the desired annotation schema, new variants were trained. The SIGMORPHON 2019 shared task is described, with winning systems described in detail. A novel architecture, `DogTag`, is proposed. All systems are implemented, and are evaluated in a similar style to the shared task, and on a novel OOV generalization benchmark. While `DogTag` is only competitive when considering the overall dataset, it proves better at generalizing to new word-forms. All findings and code is open-sourced to allow others to quickly leverage these systems.
3. **Evaluating the Morphological Awareness of NMT Systems**: this chapter proposes a method for directly testing the morphological awareness or competence of existing NMT systems. The proposed testing methodology is again novel, and clearly addresses errors or shortcomings of related methods presented in the literature. Already looking at the next chapter, the most common confused generations are found for visual analysis (e.g. singular words are produced where plurals are needed). An extensive regressions analysis is presented in the accompanying appendix, showing certain morphological features are clear determinants of poor generations.
4. **Adapting NMT Systems for Morphological Awareness**: this final chapter deals with teaching pre-trained NMT systems an inductive bias towards morphological inflection. This is done via a meta-learning framework, although the emphasis lies on regular generation of translations, not few-shot learning (i.e. the zero-shot capacity is tested). The results of the preceding chapters are incorporated to define a novel sampling scheme 'morphological cross-transfer', that disentangles for the model lemmas and affixes. While the models trained in this manner show some improvement, by the standards set out in Chapter 2, it comes at the cost of general translation competence. Some avenues for future research are suggested

Morphological Tagging and Lemmatization in Context

2

This first chapter deals with morphology as a concept, and automated analysis in natural language. The first section provides a whirlwind tour of the field, intending to provide the reader with a working notion of core definitions used throughout the remaining text. Special effort is made in elaborating differences between languages. The second section deals solely with re-implementing contextual joint lemmatizers and morphological taggers. These were deemed necessary for later efforts, but existing solutions proved inadequate. A novel architecture proves competitive with the state-of-the-art, generalizing well to out-of-vocabulary terms. No reference to machine translation or multilingual modelling is made yet, but these mono-lingual systems play a pivotal role in the coming chapters.

2.1 Morphology

Linguistically speaking, the field of morphology is the study of the smallest, most atomic units of language that carry meaning. The units, called morphemes, are present within all words, and largely define their internal structure. More specifically, the field seeks to understand these constituents of a word, their function, and whether their presence is due to grammatical or semantic necessity. Haspelmath and Sims [8] provide two succinct definitions: morphology is either the study of i) systematic covariation in the form and meaning of words, or, ii) the combination of morphemes to yield words. Not unexpectedly then, morphology is considered an important aspect of linguistics:

[8]: Haspelmath et al. (2013), *Understanding morphology*

> Morphology is the conceptual centre of linguistics. This is not because it is the dominant sub-discipline, but because morphology is the study of word structure, and words are at the interface between phonology, syntax and semantics. Spencer and Zwicky [9]

[9]: Spencer et al. (1998), *The Handbook of Morphology*

More intuitive perhaps than the notion of a morpheme is that of a word. For simplicity's sake, a word or token will be defined as some contiguous sequence of characters with some natural boundary pre- and succeeding. As alluded to earlier, a word is in fact already a compound structure. Two distinct variants of a word exist:

- when considering the abstract meaning of a word, one is considering the lexeme. Many words can belong to this lexeme, but are all represented by the same lemma. Lemmas are the items by which a dictionary is indexed, capturing some core concept, condensing a whole set of words into one
- when considering the concrete form of a word or token, one is considering the word-form, surface form or orthographic form of said word. The word-form augments the bare meaning of the lemma with morphemes that carry grammatical function, or alter the word's meaning

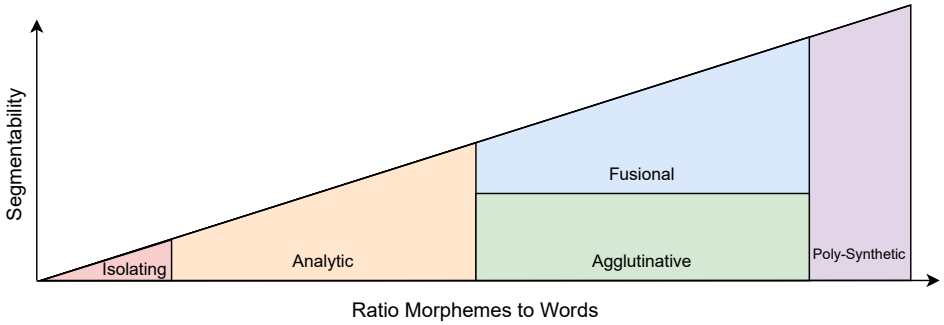


Figure 2.1: The morphological typology landscape and commonly identified subsets within.

In short, we read word-forms, and think in terms of lemmas. Consider, for example, the sentence,

> A snare is for catching rabbits; once you have caught the rabbit, forget about the snare.
> Words are for catching ideas; once you have caught the idea, forget about the words. Zhuang [10]

[10]: Zhuang (300BCE), *Zhuangzi*

In the first line, the words 'catching' and 'caught' are two word-forms referring to the same meaning, one referring to a general act, the other describing the successful completion of that act. Both are concrete instances of the lemma 'catch'. In the second line, 'caught' refers to an entirely different concept (i.e. understanding), and thus a different lexeme, but matches in word-form and maps to the same lemma.

Beyond sage advice, the example illustrates the effect of morphemes. In short, morphemes come in two forms: roots and affixes[1]. The root carries the meaning, the affixes change it to fit in the sentence. If the change is lexically motivated, i.e. to create a new vocabulary item, the word-formation process is one of derivation. If instead it is syntactically motivated, the affix is deemed functional, and it becomes an inflection. For example, to take the lemma 'catch' to 'caught', 'ught' is suffixed to the root 'ca', an inflection required due to a change in tense. All word-forms together, belonging to the same lemma constitute that lemma's paradigm (e.g. 'catch' includes in its paradigm the tense inflected forms 'catch', 'caught', 'caught').

1: Which in turn come in various flavours.

- If preceding the root, it is a prefix
- When succeeding the root, it is a suffix
- In the rare case it is placed within the root, it is an infix
- The opposite of an infix, requiring both a pre- and suffix, is the circumfix

Rules of word-formation, which come naturally to native-speakers, follow (for the most part) strict patterns. Such patterns are typically linked to a word's part-of-speech (PoS). The number and specific instances of those PoS differ somewhat between annotation schemas, but standard entries include nouns, verbs, etc. One important dichotomy between PoS are the open and closed classes, tightly linked to the notion of derivation and inflection, respectively. The former deals primarily with lexical items, and is limited only to our ability to invent new meanings. The latter, in contrast, contains words necessary to make grammatical sentences, with practically no information carried in isolation (for example, conjunctions like 'and' or 'but' or 'or').

2.1.1 Morphological Typology

While understandably important to study for each language separately, differences across languages in their approach to morphology provide a useful framework for classifying languages and specifying their relationship. This subfield is called morphological typology.

Historically, two useful metrics exist for classifying where a language falls in the morphological typology landscape [11]. The first is the degree of synthesis, defined as the number of morphemes per word. Languages with low synthesis are deemed 'analytic', whereas those with relatively high degree, 'synthetic'. For an example clearly showing the difference, compare Haspelmath and Sims [8]'s glossed translations. First an extremely analytic[2] language, Vietnamese, to a moderately analytic language like English,

2: Low synthesis are sometimes categorized as 'isolating'

Vietnamese	Hai	d-ú.a	bo?	nhau	là	ta.i	gia-d-ình	thàng	chông.
Morphemes	two	individual	leave	each other	be	because of	family	guy	husband
English	They divorced because of his family								

Next, an extremely synthetic[3] language like West Greenladic,

West-Greenlandic	Paasi-nngil-luinnar-para	ilaa-juma-sutit.
Morphemes	understand-not-completely-1SG.SBJ.3SG.OBJ.IND	come-want-2SG.PTCP
English	I didn't understand at all that you wanted me to come.	

3: High synthesis outliers are sometimes categorized as 'poly-synthetic'

In essence, the degree of synthesis dictates the allowed complexity of individual words in a sentence. For highly analytical languages, each word is typically composed of a single morpheme, and has a single semantic or syntactical function. In comparison, highly synthetic languages have words consisting of many morphemes, whose combination might allow for many distinct functions. Most major European languages tend to be synthetic, with modern English being one of the few exceptions.

The second metric of relevance is the segmentability of morphemes, called the degree of agglutination. Low levels of agglutination indicate that morphemes combine in often irregular patterns, and is typical of 'fusional' languages. Juxtaposed are the 'agglutinative' languages, having high levels of agglutination, and highly patterned composition. A prototypical example of an agglutinative language is the concatenative morphology of Turkish. Taken from Comrie [11], the following table provides the noun conjugation of the word 'walk' ('adam'), with hyphens added for effect. Note the highly predictable shift from singular to plural forms, across each casing form. Contrast this to a fusional language like Russian. This

Case/Number	**Singular**	**Plural**
Nominative	adam	adam-lar
Accusative	adam-ı	adam-lar-ı
Genitive	adam-ın	adam-lar-ın
Dative	adam-a	adam-lar-a
Locative	adam-da	adam-lar-da
Ablative	adam-dan	adam-lar-dan

table [11] instead provides a similar paradigm for the Russian nouns 'table' ('stol'), and 'lime tree' ('lipa'). Two words are included to show the

effect of a third variable of complexity interacting with the previous two, namely noun declension types. Note how the casing affixes across the singular and plural forms are often conflated, which in turn also vary across the declension type (I or II).

Declension Type	**I**		**II**	
Case/Number	**Singular**	**Plural**	**Singular**	**Plural**
Nominative	stol	stol-y	lip-a	lip-y
Accusative	stol	stol-y	lip-u	lip-y
Genitive	stol-a	stol-ov	lip-y	lip
Dative	stol-u	stol-am	lip-e	lip-am
Locative	stol-om	stol-ami	lip-oj	lip-ami
Ablative	stol-e	stol-ax	lip-e	lip-ax

Both degrees of synthesis and agglutinivity correlate positively with the notion of morphological complexity. However, the relationship is conceptually not unbounded; the distinction between fusional and agglutinative languages typically only makes sense between synthetic languages. For an overview of the morphological landscape, see Figure 2.1. When talking about morphologically rich languages, one typically refers the demarcation between synthetic and analytic languages, particularly towards poly-synthetic. Again, relative to English, most European languages tend to more complex in their morphology.

2.1.2 Morphological Annotation

Before discussing modelling approaches to morphology, a significant initial hurdle to overcome is developing a labelling scheme that is consistent across languages. At this point, the notion that languages vary drastically according to their morphological complexity should be clear. The dimensions along which that complexity is expressed are morphological features. In this too, there exists a great deal of variation.

To this end, SIGMORPHON[4] has developed the Universal Morphological Feature (UniMorph) schema [12]. Focused entirely on predicting inflected word-forms, the ultimate goal of the project is multilingual lookup of any word-form from the combination of lemma and features. Otherwise, with the lexical item known, one need only choose a single item from its paradigm to construct a grammatically correct token. This work primarily uses data annotated with UniMorph 2 [13]. Recently, the fourth version released, expanding the 23 dimensions to cover 122 million inflections across 182 languages [14]. The full schema, to which many references will be made throughout, made be found in Sylak-Glassman [12].

4: Special Interest Group on Computational Morphology and Phonology

[12]: Sylak-Glassman (2016), 'The composition and use of the universal morphological feature schema (unimorph schema)'

[12]: Sylak-Glassman (2016), 'The composition and use of the universal morphological feature schema (unimorph schema)'

2.2 Automated Morphological Tagging & Lemmatization in Context

In order to produce fluent text, one needs to choose the right words and as seen earlier, choose the right forms of those words. To do so, information from a variety of different sources needs to be combined. Word choice is not merely semantically motivated, but must be altered to conform to syntax and meaning already present within a sentence. Context is thus paramount.

From a modelling perspective, one important implication of word-formation processes are the large output vocabulary sizes. All possible lemmas are able to take on large paradigms. For morphologically rich languages, this phenomenon is especially endemic, with infinitely productive inflection systems. In turn, this leads to specific word forms or even entire morphological rules not being present in training text of morphologically rich languages, despite size. Generalization to out-of-vocabulary terms is thus crucial.

Strong automated morphological analysis systems must thus be able to quickly infer, from context, which features are present for any word, be able to separate those features in word-form space from the underlying lexical item, and ideally do so from limited data, all the while retaining the ability to be infinitely productive.

Running from December 2018 till August 2019, this is precisely the main focus of the second 2019 CoNLL-SIGMORPHON Shared Task McCarthy et al. [15]. Where previous renditions and other tasks focus on automated parsing of paradigms under a variety of constraints, this was the first (and since only) shared task aimed at incorporating contextual information for full sentences. Specifically, for all tokens present in a string, models are expected to produce i) its lemma, ii) the part-of-speech and iii) the morphological features. Due to the task's setup, large neural systems lend themselves especially well.

While unique to SIGMORPHON's workshops and shared tasks, the requirements are similar to the shared tasks presented by SIGNLL[5]'s CoNLL[6] conference. Specifically, the labelled corpora necessary for supervised learning are present in the Universal Dependencies (UD) project. At the time (version 2.3), the project served as a repository of 129 pre-tokenized treebanks across 79 languages. In turn, the languages are spread across a wide array of typological families, with a wide variety of lower resource languages being included. All languages share an annotation scheme, in theory, and are marked for quality to distinguish the annotation's reliability. Furthermore, all treebanks are provided in official train/dev/test splits, enabling robust comparison across systems. Using an automated modification process, the SIGMORPHON shared task requires predicting 2 out of 11 provided labels for all tokens in the treebanks, and are evaluated on the test-set performance averaged over all provided treebanks.

Table 2.1: An annotated sentence from the Dutch LassySmall treebank, showing the tokenized text and the labels to predict.

Token	Lemma	Feats.
In	in	ADP
1425	1425	NUM
ging	gaan	V;SG;PST;FIN
hij	hij	3;PRO;NOM
naar	naar	ADP
Rijsel	Rijsel	SG;PROPN;NEUT
,	,	PUNCT
waar	waar	ADV
hij	hij	3;PRO;NOM
hof-	-	-
schilder	schilder	N;SG;MASC+FEM
⋮	⋮	⋮

[15]: McCarthy et al. (2019), 'The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection'

5: Special Interest Group on Natural Language Learning

6: Conference on Computational Natural Language Learning

2.2.1 Lemmatization as Classification

A clear division in the submitted systems comes from viewing the lemmatization task as either character-based seq2seq generation or token-level classification. With all systems relying on either a recurrent or self-attention based architecture, moving towards an encoder-decoder setup is a natural extension. By limiting the decoder's output vocabulary to a language's alphabet, the system is infinitely productive with few necessary parameters in the final classification head, but suffers in computational complexity with a greatly increased number of units to classify. A common alternative that retains the benefits of an encoder-decoder setup is predicting an edit operation, or set of edit operations, instead of individual characters. Consecutive edit operations can be concatenated together, requiring an additional label but yielding fewer total number of classifications.

Ultimately, whether characters or character edits, seq2seq generations of lemmas from word-forms requires successful transfer of word-based context to a character-based decoder. The potentially long character sequences and reduced representational power of the decoder can lead to a bottleneck, or at least severely complicates training and prolongs convergence. A far simpler, but more restrictive, method is predicting entire edit-scripts (a concatenation of all character-based edit actions for the entire token). Perhaps an engineering necessity, this recasting of lemmatization as classification proves an effective simplification, being utilized by both winning systems[7]

First proposed as an automated pre-processing step by Chrupala [16], lemmatization as multi-class classification requires finding for tokens in the train set a minimal or shortest edit script between the lemma and word-form. This script consists of a number of specific operations, yielding a deterministic mapping from a token's word-form to its lemma. Actions, defined at the character level, are typically restricted to skipping ("*"), deletion ("-") or insertion ("+?"), with the last action requiring a specific form for all possible insertion symbols. From the two sequences, and specifically these allowed operations, the classic Myers difference algorithm may be applied [17], finding the minimal edit script by searching the edit graph for the shortest path solution, finding application in (for example) Git's diff function [18].

Going from a shortest edit script between two strings to a lemma edit script between a word-form and lemma representation of the same string, requires a few additional steps. First, the longest common sub-string is found, is deemed the stem, and removed from further consideration. All text preceding the stem is considered a prefix, and all text succeeding is a suffix. For both affixes, the case insensitive shortest edit script is found with the operations detailed above. Finally, for the entire string, the occurrences of capitalized characters are noted, with sequences being compressed to their first character. The concatenation of all three components (casing, prefixes, affixes) is the lemma edit script. As a pre-processing step, this is performed for all tokens present in the dataset, enumerated and converted to a one-hot encoded multiclass label vector. One important special case is when no common sub-string is found. They are considered irregular, and given an edit script that ignores the provided token altogether.

Figure 2.2: An example of a lemma edit script.

7: Compared to systems predicting edit actions, the nearest generative system (CBNU) pre-trains a tiny-transformer exclusively for lemmatization. This yielded a lemmatizer only marginally better than the shared task's baseline, and a morphological tagger significantly worse than the winning systems.

[16]: Chrupala (2006), 'Simple data-driven context-sensitive lemmatization'

[17]: Myers (1986), 'An O(ND) difference algorithm and its variations'

[18]: Coglan (2020), *Building Git*

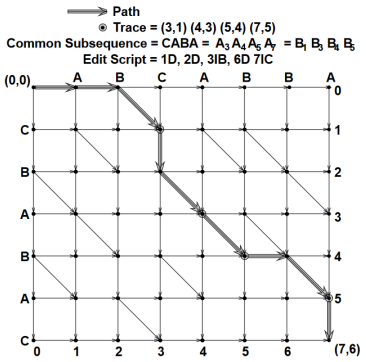


Figure 2.3: The shortest edit script is the shortest path across the edit graph defined on the tokens of both sequences. Taken from [17].

Rule	Count	Examples
L0 | d | d	450024	i→i, the→the, like→like
L0 | d | —	35460	flights→flight, arrives→arrive, later→late
U0,L1 | d | d	27682	President→President, Tuesday→Tuesday
L0 | ign_be	11628	am→be, is→be, ‘m→be
L0 | d | ——	10252	sixth→six, does→do
U0 | d | d	8944	i→I, I→I, AP→AP
L0 | d | ———	6917	returning→return, cheapest→cheap
L0 | —+b | d	3321	Are→be, ‘re→be, are→be
L0 | d | ———+e	3295	making→make, leaving→leave
L0 | d | —+v+e	2841	has→have, had→have, HAS→have
U0 | ign_I	2745	me→I, Me→I, my→I
L0 | d | ———+y	2039	cities→city, earlier→early, carries→carry
L0 | d | ——+e	1932	ninth→nine, his→he, him→he
L0 | d | —+o*	1646	grew→grow, knew→know, n’t→not
L0 | d | ——+y	1433	paid→pay, said→say, their→they

Table 2.2: The 15 most common lemma edit scripts for English treebanks. The first column gives the script, the second the frequency of the script and the final column some examples from the corpus, as token → lemma.

Overall, while not productive, using lemma edit scripts as labels strikes a nice balance between compressing the lemma space and generalizing to unseen word-forms. Furthermore, the methodology is language agnostic, depending only on the tokenizer. The number of tokens corresponding to a lemma edit script follows a power law, with the vast majority being captured using a small portion of the most common scripts. The most frequent edit scripts tend to be those where the token is already close to the lemma, requiring few to no actions. Consequently, the least frequent edit scripts tend to be for longer words, or ones containing rare character combinations in the affixes.

2.2.2 Architectures

Presented in this subsection are the setups of the 2 winning systems, and a novel architecture leveraging a more recent character-based transformer. All architectures classify a lemma script and a morphological tagset, jointly, for all tokens in a sentence.

UFAL-Prague’s `UDPipe2`

As the name suggests, `UDPipe2` [19] is the second iteration of a recurrent multitask NLP pipeline. In its standard form, it is trained to simultaneously part-of-speech tag, lemmatize and parse dependency relationships between tokens. This setup proved state-of-the-art for a previous CoNLL shared task [20], and needed few modifications for the SIGMORPHON/-CoNLL 2019 shared tasks. At it core, it uses a standard NLP setup. Words are fed through a variety of embeddings, which are passed on to a deep bidirectional recurrent block, before being classified with task-specific multilayer perceptrons (MLPs).

The word-level embeddings consist of three forms: pre-trained language-specific subword aware `fastText` embeddings [21, 22]; contextualized embeddings from feeding the entire sentence into `BERT` [23] and recovering tokens from the averaged BPE representation of the final four layers; and finally, trainable token-specific word embeddings. A fourth type of

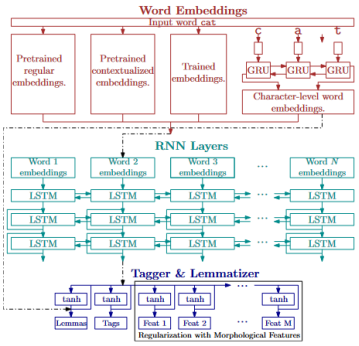


Figure 2.4: The `UDPipe2` architecture presented graphically. Taken from [19].

[19]: Straka et al. (2019), ‘UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging’

[20]: Straka (2018), ‘UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task’

'word embeddings' are provided in the form of a trainable `char2word` module [24]; tokens are fed as characters into a bidirectional `GRU` [25], with directions being concatenated and projected down before sum-pooling over the time dimension. In turn, the recurrent block consists of a 3 layer bidirectional `LSTM` [6], with skip-connections between the layers [26]. The final block consists of 2 layer MLPs, again with skip-connections, each classifying the produced logits for a separate task. The char2word embeddings are appended to the contextualized token representations prior to lemmatization, adding another skip-connection. As a method of regularization, the morphological tags are classified jointly, but also factored into categories, with the latter loss only being used during training. Regularization proved crucial for generalizability, with dropout being applied throughout and label smoothing when classifying.

`UDPipe2` relies heavily on pre-trained neural modules, at a variety of context levels. These are all simply concatenated before re-contextualizing. As a result, this architecture has proven useful to a number of NLP tasks, with more relevant embeddings being quickly slotted in, as evidenced by Straka and Straková [27], where swapping out `BERT` for `RoBERTa` [28] provided a modest performance bump. Thus, despite having many parameters, relatively few are trainable for the task at hand, while most were previously exposed to large amounts of data with more general objectives. This makes re-training comparatively efficient, and quickly allows for scaling to larger datasets, at the cost of an increased memory footprint and inference latency.

[27]: Straka et al. (2020), 'UDPipe at EvaLatin 2020: Contextualized embeddings and treebank embeddings'

Charles-Saarland's `UDIFY`

Kondratyuk presents `UDIFY` [29, 30] as a system very similar to `UDPipe2`. Instead of relying on 4 somewhat similar word-embedders, only `BERT` and char2word are kept. Furthermore, `BERT` is made trainable, with token embeddings being created by attending over all self-attention layers, keeping only the first BPE. Where `UDPipe2` uses a tightly joined recurrent block, `UDIFY` separates these into two smaller `LSTM`s: one for morphological tagging and one for lemmatization. Otherwise, differences are minor at best.

Previous research into transformer-based architectures have already indicated the highly hierarchical representations built by `BERT` and the like; lower layers tend to specialize in classically upstream tasks, whereas upper layer representations contain more upstream, context-dependent information [31]. Layer attention, a simple static vector of linear mixing coefficients, as utilized by `UDIFY` could lead to automatically detecting and leveraging these specializations. This effect is then compounded by passing gradients back to the decoder-only block. While consisting of relatively many trainable parameters, and sacrificing the latency gains of self-attention modules by introducing recurrent blocks throughout, `UDIFY` remains fully language agnostic, with the individual components all being shown to scale to multilingual settings. Kondratyuk leverages this property well, with optimal performance only being achieved when pre-training on all available languages, before fine-tuning to each treebank in isolation.

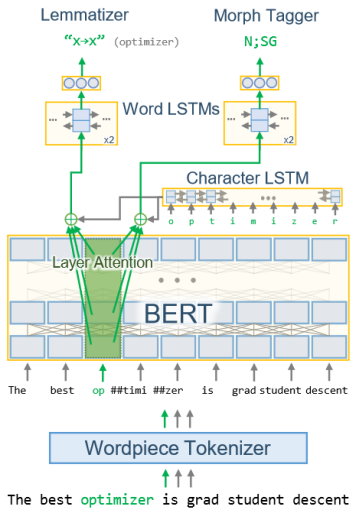


Figure 2.5: The `UDIFY` architecture presented graphically. Taken from [29].

[29]: Kondratyuk (2019), 'Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning'

[30]: Kondratyuk et al. (2019), '75 languages, 1 model: Parsing universal dependencies universally'

[31]: Tenney et al. (2019), 'BERT rediscovers the classical NLP pipeline'

DogTag

In the years since, transformer architectures have gone from wildly impressive newcomers to virtually ubiquitous, with improvements being made along the way. While the hidden states contain some useful representations for lemmatization and annotation, they remain largely unaware of morphological features in word-form space, due to commonly used sub-word tokenizers. One specific architecture that proves an exception to this rule is Clark et al.'s `CANINE`[32]. Using a series of convolutions for down- and up-sampling, they manage to train a deep transformer stack on character input and output. Much like `BERT`, the first token of the transformer stack contains a semantic representation of the entire sentence, but only the upsampled character representations are used for the masked language modelling objective. Via a relatively simple added module, they produce a model that is entirely tokenizer free, vocabulary free, almost inherently multilingual and capable of handling long sequences - all with 30% fewer parameters than sub-word alternatives[8]. However, while provably impressive at natural language understanding tasks, due to the character based MLM objective, the final layer is by necessity aware of morphological word-formation processes. Hence, the final layer representations likely contain information pertinent to tasks like lemmatization and morphological feature prediction.

[32]: Clark et al. (2022), 'Canine: Pre-training an efficient tokenization-free encoder for language representation'

`DogTag` follows this line of reasoning: a joint lemmatization and morphological feature prediction model in the style of `UDPipe2` or `UDIFY`, but with `CANINE` as the sole feature extractor. The contextualized character representations produced by `CANINE` are collated using multi-head attention [2] between the character sequence and a learnable query matrix (of length 1). Beyond the multiple heads (whose number correlated positively with performance), three separate character collators are trained: two for directly feeding into lemmatization and morphological feature prediction, and one being fed into a bidirectional, multi-layer residual `LSTM` for recontextualization of the token representations. These representations are then concatenated with the task specific collations. A simple `MLP` is trained to classify tokens, with training only regularization coming from a factored prediction. While initial experiments keep `CANINE` fixed, fine-tuning the feature extractor proved beneficial, although unstable at times.

8: However, given the auto-regressive modelling of characters, the authors do show a 50% increased sentence throughput.

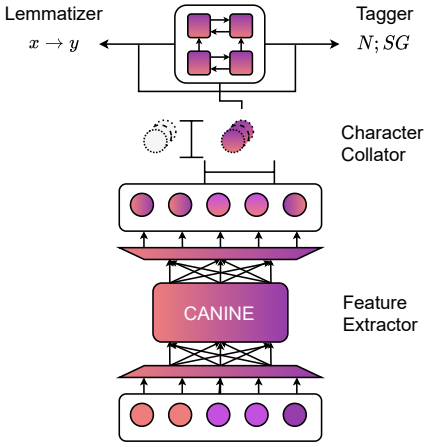


Figure 2.6: The `DogTag` architecture presented graphically.

2.2.3 Methods

Data

Since the 2019 CoNLL-SIGMORPHON Shared Task, the UD treebanks have seen numerous revisions and updates, both in languages used and novel. Specifically, the used version stems from 2.3, with version 2.9 and 2.10 being made available in November 2021 and May 2022 respectively. One particularly relevant change are differing train/test splits, making direct comparisons practically impossible. For many treebanks, the annotations have been brought to align more closely to the UD standard.

To leverage the improved data quality, treebanks from version 2.9 were used. The `UPOS` and `XPOS`, carrying the Part-of-Speech (PoS) and morphological feature tag sets respectively, were converted to Universal

Table 2.3: Differences between UD and UM annotations. Taken from [33].

Schema	Annotation
UD	VERB MOOD=IND | NUMBER=SING | PERSON=3 | TENSE=IMP | VERBFORM=FIN
UniMorph	V;IND;PST;3;SG;IPFV

Morphology (UM) [13] tagsets using the same methodology as presented by McCarthy et al. [15]. The work by McCarthy et al. [33] presents an automated conversion system that merges PoS with morphological features, with language specific combinations being considered, leading to increased cross-lingual agreement. They further show improved tagging recall scores compared to simply using the UD features. Lemmas were left intact, despite systematic differences across tree-banks, and all other provide features were left in CoNLL-U format. For an extended overview and explanation of all UniMorph morphological tags, see Sylak-Glassman [12].

[33]: McCarthy et al. (2018), 'Marrying Universal Dependencies and Universal Morphology'

[12]: Sylak-Glassman (2016), 'The composition and use of the universal morphological feature schema (unimorph schema)'

Unlike the shared task, this re-implementation focuses on language-level training, concatenating available treebanks. One unavoidable source of systematic errors are differences in annotation methodologies across different treebanks. While the UD project has prescribed a standard, ultimately their primary function is collating available treebanks, leaving manual annotation to separate authors. This leaves mixtures of treebanks, and in turn entire languages, riddled with noisy labels. Some treebanks especially do not lend themselves to the task of automated morphological tagging and lemmatization due to incomplete or unconventional schemas. Therefore, UD provides quality ratings on their website, loosely based on the unattached undirected attachment score (UUAS) of the provided treebanks, with scores close to 0 indicating particularly poor treebanks. To avoid inclusion of harmful treebanks, a lower quality limit of 0.2 (1 out of 5 stars) is adhered to. For an overview of available language corpora, their size and provenance, see Appendix A Table A.1.

Experimenting with all available languages is prohibitively expensive. Instead, 8 (primarily Indo-European) languages were chosen for a good mixture of typological families, morphologically complexity. See Table 2.4.

Table 2.4: Chosen languages. Columns give their typological families, the language name and their position on the morphological spectrum (Analytic (Ana.), Synthetic (Syn.), Fusional (Fus.) and Agglutinative (Agg.).

Fam.	Lang.	Type
Germanic	Dutch	Syn.
	English	Ana.
Romance	French	Fus.
Slavic	Czech	Fus.
	Russian	Fus.
Semitic	Arabic	Fus.
Turkic	Turkish	Agg.
Uralic	Finnish	Agg.

Implementation Differences & Hyperparameters

Given new datasets and a slightly altered end-goal, namely strong language-specific morphological taggers for a downstream task, the use of intention behind using pre-defined architectures is not replication of earlier results, but merely re-implementation of strong baselines. Furthermore, the cited systems were each developed using NLP-specific deep learning frameworks, whereas their re-implemented versions are built using PyTorch [34] only. Some implementation details differ:

- **Factoring**: both `UDPipe2` and `UDIFY` use factored (i.e. the morphological tags split into separate categories) tag sets as a training-only form of regularization. With the new datasets, this proved difficult to re-implement, as some tags were not linked to any category, and some to multiple. Instead, models were trained to classify present morphological categories along side all tags.
- **Sparse Embeddings**: support for sparse embeddings remains lacking in PyTorch, and lead to instabilities. Instead, for embedding layers, back-propagated gradients were dense with a high β_2 value (0.999) for the Adam optimizer [35] instead.
- **Additional Regularization**: the most prevalent issue in test-set performance appears to be overfit to the training dataset. While

both cited architectures use high levels of dropout throughout, additional regularization techniques proved beneficial. Of note are masking of entire words and characters when feeding input sequences to pre-trained embedders

- **Competition tricks**: rather than focusing on building systems capable of winning on a per-treebank basis, the systems are designed for performance on a single language. This invalidates some training strategies employed by the original authors. For example, no additional per-treebank fine tuning is done, nor is ensembling of systems, nor language specific hyperparameter searches.

Despite these alterations and augmentations, the hyperparameter sets used were taken directly from the original papers. Overall, the systems proved reasonably robust to most choices, and no language-specific changes were introduced. All results and corresponding hyperparameter choices may also be found in the Weights & Biases [36] dashboard used for experiment tracking[9]. Used code, datasets and model checkpoints has been documented and open-sourced [10], and should allow for easy replication.

9: https://wandb.ai/verhivo/morph_-tag_lemmatize

10: https://github.com/IvoOVerhoeven/morph_-tag_lemmatize

2.2.4 Results

Overall Test Set Performance

In the style of the shared task, overall test set performance metrics are provided in Table 2.5. The values are computed using all available sentences in the official test splits of the used datasets. Again, direct comparison is not possible, but these do provide comparisons between systems, and ballpark estimates of how these systems compare to their original counterparts. For `UDPipe2` and `UDIFY` the self-reported metrics are provided, averaged over the languages included here. System wide means are also provided, averaged over all used languages[11].

11: This practise has come under scrutiny, with high-resource language families unfairly inflating average scores. Averaging over typological family averages can change the systems' rankings [37].

Both `UDPipe2` and `UDIFY` are clearly strong baselines, extending their impressive competition results to updated datasets and a slightly altered training procedure. Give `UDPipe2`'s heavy reliance on pre-trained resources to provide morphologically or context aware word embeddings, the model requires only a fraction of the training time used by its rivals, making its performance all the more striking. In the officially published results, it already achieved the best lemmatization performance, and was the second best tagger, with notable improvement from corpora merging. Here, it's primarily the morphological tagging performance that stands out. Overall, however, `UDIFY` appears to be the winner. It sets the system-wide highest metrics for most languages, and does so with a respectable margin. Strangely enough, on some of the higher resource languages its performance falters relative to the rest. As such, its across language performance is on par with `UDPipe2`.

Comparatively, `DogTag` uses only a fraction of the parameters, and in the `DogTag`-Fixed variant, trains only a fraction of that fraction. Allowing the fine-tuning of the `CANINE` backbone proves important, bringing the system to close to SoTA. Even without the fine-tuning, `DogTag` proves a strong lemmatizer especially. Overall, it seems the best lemmatizer, and

Table 2.5: Test set performance of `UDPipe2` and `DogTag` (with pre-trained `CANINE` weights, and mono-lingual finetuned variants). Metrics are provided per-language, and mean aggregated per system in the MEAN row. All standard deviations were below 5e-3, and are omitted for brevity's sake. Numeric columns provide, in order, the 0-1 accuracy of a tokens predicted lemma, the Levenshtein distance between predicted and ground-truth lemma, the 0-1 set accuracy of tokens predicted morphological feature set, the F1 score for morphological tags (presented as micro/macro averaged), and finally the throughput in tokens per second, as measured on a NVIDIA GTX 1080Ti GPU, with batches of 2048 tokens and at most 248 sentences. Bold values indicate best across systems. Arrows ↑↓ indicate whether higher or lower values are desired, respectively. For `UDPipe` and `UDIFY` the self-reported competition results are presented as a *rough* baseline. For `UDIFY`, the multilingual with mono-lingual fine-tuning models are used as baselines. Given the differences in training, and the different datasets, direct comparison is not recommended. Instead, these present an upper limit to the performance of the replicated models.

Model	Language	Lemma Acc. ↑	Lev. Dist. ↓	Morph. Set Acc. ↑	Morph. Tag F1 ↑	Throughput ↓
`UDPipe2`	Arabic	0.93	0.21	0.90	0.96/0.85	**2313**
	Czech	0.98	0.03	0.92	0.98/0.90	**2930**
	Dutch	0.94	0.12	0.95	0.97/0.93	**3223**
	English	**0.97**	**0.05**	**0.92**	**0.96/0.90**	**2977**
	Finnish	0.82	0.44	0.81	0.92/0.62	**2633**
	French	**0.98**	**0.04**	**0.92**	**0.97**/0.87	**3716**
	Russian	**0.97**	**0.06**	**0.92**	**0.97/0.88**	**2759**
	Turkish	0.91	0.19	0.77	0.89/0.58	**1828**
	MEAN	**0.94**	0.14	**0.89**	**0.95**/0.82	**2797**
	[19]	0.96	0.11	0.95	0.98/	–
`UDIFY` Mono	Arabic	**0.94**	**0.18**	**0.93**	**0.96/0.88**	2135
	Czech	**0.99**	**0.02**	**0.95**	**0.98/0.95**	2413
	Dutch	**0.95**	**0.09**	**0.96**	**0.97/0.96**	2507
	English	0.93	0.12	0.82	0.89/0.82	2445
	Finnish	**0.88**	**0.26**	**0.92**	**0.96/0.84**	2106
	French	0.94	0.18	0.93	0.96/0.88	2135
	Russian	0.92	0.14	0.77	0.91/0.80	2242
	Turkish	**0.94**	**0.13**	**0.83**	**0.92/0.74**	1371
	MEAN	**0.94**	0.14	**0.89**	0.94/**0.86**	2169
	[29]	0.95	0.11	0.95	0.98/	–
`DogTag` Fixed	Arabic	0.85	0.45	0.76	0.90/0.77	1851
	Czech	0.93	0.12	0.72	0.90/0.81	2255
	Dutch	0.87	0.27	0.79	0.87/0.83	2255
	English	0.93	0.12	0.79	0.88/0.81	2361
	Finnish	0.67	0.85	0.55	0.77/0.52	1620
	French	0.95	0.09	0.84	0.93/0.81	1652
	Russian	0.91	0.16	0.73	0.89/0.78	2110
	Turkish	0.79	0.46	0.60	0.81/0.53	1714
	MEAN	0.86	0.32	0.72	0.87/0.73	1977
`DogTag` Mono	Arabic	0.93	0.20	0.87	0.94/0.84	1851
	Czech	0.98	0.03	0.90	0.97/0.93	2279
	Dutch	0.93	0.13	0.92	0.95/0.94	2256
	English	**0.97**	0.06	0.89	0.94/0.87	2282
	Finnish	0.85	0.32	0.85	0.94/0.81	1915
	French	**0.98**	**0.04**	**0.92**	**0.97/0.88**	2623
	Russian	**0.97**	**0.06**	0.88	0.96/**0.90**	2135
	Turkish	0.92	0.18	0.78	0.91/0.72	1714
	MEAN	**0.94**	**0.13**	0.88	**0.95/0.86**	2132

while it lags on morph tag set accuracy, the F1 scores stand out. This might indicate slightly better capacity at handling rare features.

There exist notable differences within languages, but across systems. The three smallest corpora, Arabic, Finnish and Turkish[12], yield lower lemmatization scores, and substantially lower morphological tagging results. For Finnish and Turkish especially, the macro averaged F1 scores lag behind (behind the '/'). Again, low values on especially this metric are symptomatic of not being able to handle infrequent label instances.

12: Interestingly, these also represent the included agglutinative languages.

More disappointing was the lack of improvement from the 2 stage multi-lingual pre-training followed by mono-lingual fine-tuning setup, as prescribed by `UDIFY`. Results may be found in Appendix A Table A.2. The `UDIFY` shows definite improvement, approaching the self-reported global averages. Relative to their monolingual training, however, the improvement is smaller. It could be that treebank merging already provides much of the benefit that including other languages from the same typological family provides. Otherwise, the system might be close to a feasible upper limit on performance. For `DogTag`, hardly any improvement was booked overall, and the Turkish system even showed decline. Potential explanations might include a lack of consistency in the language merged labels, or requiring merging of the dataset beyond the typological family.

Generalization to Out-of-vocabulary Terms

While the use of hidden test set does emulate the system's performance on a natural language corpus of the same language, it does not directly test the generalizability of the systems to new word-forms and new lemmas. Especially for morphologically complex languages, this ability is a necessity for downstream use. Thus, moving beyond the analysis provided by the task organizers, the generalizability of the `UDIFY` and `DogTag` are put to the test. Two forms of generalization are tested for, i) comparing seen word-forms to unseen word-forms of known lemmas, and ii) comparing seen lemmas to unseen ones. Both test to which degree the models can leverage provided information beyond their capacity to memorize word-forms. Such a test ideally checks both the lemmatization and morphological tagging capacity. The chosen metrics, one for each task, are the Levenshtein distance and the intersection over union (IoU) of the produced tag set. The latter is a more fine-grained measure of token-level accuracy than presented in Table 2.5.

Table 2.6 displays the results per-language, averaged. The mean generalization gap between seen and unseen is provided in terms of the within group variance via Cohen's d[13], with ideal values being 0 (no difference between seen and unseen). To test the differences systems, p is provided, indicating the one-sided probability that the d value for `UDIFY` and `DogTag` are statistically significant[14].

The Levenshtein distance between predicted and ground-truth lemmas shows a counter-intuitive pattern when comparing generalization to new word-forms and to new lemmas. In the former situation, only the particular inflection is unfamiliar. One would expect a successful lemmatizer to be able to disentangle the affix from a lemma, with affixes generally being shared across paradigms. However, when comparing

13: Rather than reporting a standard statistic for difference of means, like Student's or Welch's t-test, Cohen's d [38] is used; the difference in means in units of the pooled group standard deviation. Unlike the aforementioned statistics, Cohen's d provides sensible values when variances between groups differ and populations are of unequal sizes. This is the case for comparing seen word-forms/lemmas to unseen ones. The former is far more frequent than the latter. Cohen's d has as sampling distribution a standard normal distribution, and the variance is approximately known [39].

14: While hypothesis testing is hinted at, please note that p here is not used as a formal hypothesis test. Instead, it merely provides an indication that the observed effect is actually present. In other words, it indicates the probability that the same test repeated with a new set split would yield $d_{\texttt{DogTag}} - d_{\texttt{UDIFY}} = 0$

Table 2.6: Testing the generalization capacity of the best systems from Table 2.5. Given are, per-language and per-system, the average metric value for tokens that fall in the seen and unseen categories. Arrows ↑↓ indicate whether higher or lower values are desired, respectively, and a ⋄ indicates 0 is ideal. The difference between these is given in units of the pooled standard deviation, a statistic known as Cohen's *d*. The *p* value provides the probability that the difference $d_{\texttt{DogTag}} - d_{\texttt{UDIFY}}$ being positive is an artefact of the variance inherent to each values, with * indicating $p < 5e-2$, a standard hypothesis testing acceptance rate.

		Lev. Distance ↓							**Morph. IoU ↑**						
		UDIFY			DogTag			*p*	UDIFY			DogTag			*p*
	Lang.	**Seen**	**Unseen**	*d*⋄	**Seen**	**Unseen**	*d*⋄		**Seen**	**Unseen**	*d*⋄	**Seen**	**Unseen**	*d*⋄	
Word-form	Arabic	0.10	1.51	−2.01	0.11	1.74	−2.17	1.000	0.97	0.90	0.44	0.95	0.81	0.70	1.000
	Czech	0.01	0.13	−0.56	0.02	0.16	−0.54	0.099	0.98	0.94	0.43	0.96	0.91	0.41	0.090
	Dutch	0.04	0.58	−1.37	0.06	0.65	−1.18	0.001*	0.98	0.90	0.55	0.96	0.79	0.80	1.000
	English	0.09	0.99	−1.81	0.05	0.51	−1.13	0.000*	0.88	0.50	1.27	0.93	0.76	0.71	0.000*
	Finnish	0.06	0.69	−0.91	0.09	0.77	−0.85	0.021*	0.97	0.89	0.43	0.94	0.84	0.43	0.466
	French	0.03	0.29	−0.84	0.04	0.35	−0.92	0.914	0.97	0.92	0.38	0.96	0.88	0.47	0.946
	Russian	0.10	0.61	−0.85	0.05	0.29	−0.56	0.000*	0.90	0.77	0.54	0.95	0.91	0.22	0.000*
	Turkish	0.09	0.39	−0.51	0.13	0.56	−0.61	1.000	0.91	0.82	0.32	0.88	0.79	0.31	0.299
	Mean	0.06	0.65	−1.11	0.07	0.63	−1.00	-	0.95	0.83	0.55	0.94	0.84	0.50	-
Lemma	Arabic	0.14	1.23	−1.34	0.17	1.34	−1.37	0.686	0.96	0.75	1.23	0.94	0.70	1.15	0.050*
	Czech	0.01	0.15	−0.61	0.03	0.17	−0.53	0.000*	0.98	0.90	0.79	0.96	0.85	0.76	0.099
	Dutch	0.05	0.57	−0.98	0.08	0.64	−0.89	0.007*	0.98	0.87	0.64	0.95	0.80	0.64	0.562
	English	0.10	0.55	−0.79	0.06	0.23	−0.39	0.000*	0.87	0.62	0.81	0.93	0.80	0.50	0.000*
	Finnish	0.15	1.11	−1.10	0.19	1.08	−0.95	0.000*	0.96	0.91	0.28	0.93	0.86	0.27	0.381
	French	0.03	0.16	−0.40	0.04	0.16	−0.35	0.140	0.97	0.84	0.83	0.96	0.82	0.76	0.073
	Russian	0.13	0.50	−0.60	0.06	0.28	−0.51	0.000*	0.89	0.69	0.81	0.94	0.81	0.72	0.000*
	Turkish	0.11	0.92	−1.30	0.16	1.02	−1.18	0.001*	0.90	0.75	0.56	0.87	0.71	0.55	0.400
	Mean	0.09	0.65	−0.89	0.10	0.62	−0.77	-	0.94	0.79	0.74	0.94	0.79	0.67	-

the mean unseen distances, it becomes readily apparent that word-form generalization yields worse lemmatization performance. Two possible explanations for this effect are:

1. unseen lemmas are easier to inflect or belong to classes that are. For example, no training set could contain all possible proper nouns, but these tend to inflect in a rigid pattern. Phrased otherwise, new lexical items are likely to come from the open class of words, indicating words carrying predominantly semantic information, and tend to be inflected according to learned patterns
2. this is an artefact of limiting the lemma generation to existing lemma scripts. During training, the model is encouraged to associate a word-form to a set of classes, and neglect all scripts not immediately relevant. When presented with a new word-form, the system only chooses a lemma edit script associated with similar words in the training data

The only two languages where is not the case, Finnish and Turkish, are agglutinative. It could be that the prototypical easy segmentability present in this type of languages is at play, although currently evidence is too weak to draw any concrete conclusions. When considering the morphological tagging performance metrics, this relationship is not present, with word-form generalization scoring higher for some languages, and lower for others.

Regardless, DogTag's lemmatization proves to be more robust to OOV terms, both in the sense of surface-forms of words it has already seen,

and altogether new lexical items. While its performance lags for most languages, the difference in performance does not, implying that a general increase to DogTag's lemmatization capacity should yield better performance on unseen word-forms and lemmas also. This effect also extends to morphological tagging, although to a lesser degree. It is already clear that UDIFY is the better morphological tagger overall (see Table. 2.5)), and this extends to unseen word-forms and lemmas also. When comparing generalizbility, DogTag appears to be marginally better for both forms of generalization. While some differences exist between languages, these match the performance differences already noted earlier.

Overall, while possessing far fewer parameters, DogTag shows it is capable of leveraging character-level information in order to generalize as well as or better than UDIFY.

2.3 Discussion

By this point, a notion of what morphological word-formation processes are and how these differ across languages, should have been made clear. More importantly, modelling these linguistic phenomena using dedicated architectures is covered in detail. These systems are evaluated both in terms of general test-set performance, but also on their capacity to lemmatize and annotate novel word-forms and lexical items. All trained systems, both ones re-implemented and novel, prove their competence on both these tasks, with strong performance across a wide variety of languages.

To a lesser extent, this information was already available after the CoNLL/SIGMORPHON 2019 shared task, and is verified to function on general language corpora without the context of a competition. These already make such systems valuable tools for researchers, as will be evidenced in the later chapters of this thesis. However, despite little additional research post-competition, this does not imply this task is 'solved'. While for many languages scores appear to be approaching an upper limit on performance, lower-resource or morphologically rich (and especially both) languages lag behind, considerably. The multilingual pre-training prescribed by UDIFY could improve this facet especially, and with large pre-trained transformers forming the modelling backbone for all considered languages, larger and more varied data might narrow this gap. As done with DogTag or DogTag in later iterations, replacing the transformer backbone with newer, more morphologically aware self-attention architectures, might already yield a better inductive bias. Especially when using character-based transformers, one could step away from lemmatization by classification, rephrasing it as another seq2seq task and yielding systems better at handling open vocabularies. Plenty of additional future research presents itself. For example, typological property prediction during multilingual training has shown a beneficial effect on UDIFY [40]. In summary, while already impressive, new techniques from other NLP domains should be implemented for automated morphological tagging and lemmatization also, likely bridging the gap between higher and lower resource languages.

Evaluating the Morphological Awareness of NMT Systems

3

With strong morphological taggers and lemmatizers, adherent to the UniMorph schema, now in place, this chapter presents a specific application. While never made explicitly aware of a language's morphology, neural machine translation systems must nonetheless pick up on word-formation processes in order to become capable translators. However, in morphologically rich languages, as discussed earlier, data sparsity can lead to novel word-forms being required, requiring the model to leverage its understanding of sub-word units to recognize and generate said word-forms. The joint taggers and lemmatizers, having had this information made explicit, can be used to assess to which degree NMT systems manage this.

Related work is presented in detail, with special attention applied to their suitability for this task. Then, a novel data collection scheme is provided, and later applied. The results presented in this chapter and accompanying appendix, while useful and relevant in and of their own right, were initially only intended as a stepping stone to building samplers required for the next chapter. As such, the analysis presented in the main text is mostly oriented to its downstream task, whereas those presented in Appendix B lies closer in spirit to prior works.

A basic understanding of neural machine translation systems, especially modern transformer based architectures (see for example, Bahdanau, Cho, and Bengio [1] and Vaswani et al. [2]), and commonly used tokenizers is expected, but not crucial for understanding.

3.1 Related Work

Morphological complexity is generally considered a relevant predictor of errors, with a variety of tasks becoming more difficult with richer morphological processes. Despite this, the degree to which this is the case remains an open question. Wanting to put empirically sound findings, Belinkov et al. [41] train diagnostic classifiers, avant la lettre, to infer where and how strongly morphological features are encoded in a recurrent architecture. They find that morphology is especially prevalent in lower levels, positing that these focus on word structure. Generally, morphological complexity negatively correlates with translation performance, and even find that translating into morphologically poor languages improves the encoder's capacity to carry source-side morphological features.

[41]: Belinkov et al. (2017), 'What do neural machine translation models learn about morphology?'

Expanding on these findings, Bisazza and Tump [42], use a more fine-grained feature profile, and note that the encoder only captures morphological information if it is directly relevant (i.e. a good predictor) for target-side translation. The noted effect is especially prevalent for grammatical categories, like gender. They label the encoder as 'lazy', prioritizing passing semantic information and markers to the decoder.

[42]: Bisazza et al. (2018), 'The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation'

Base&Variant(s)	Output	Result
	A-set	
I am hungry	mám hlad	
I am not hungry	**nemám** hlad	negation found
	B-set	
I see him	vidím ho	noun and adjective both
I see a crazy researcher	vidím **bláznivého výzkumníka**	have accusative form
	C-set	
I agree with the president	souhlasím s **prezidentem**	all nouns bear
I agree with the director	souhlasím s **ředitelem**	the same
I agree with the minister	souhlasím s **ministrem**	intrumental case
I agree with the driver	souhlasím s **řidičem**	
I agree with the painter	souhlasím s **malířem**	(Entropy = 0.0)

Figure 3.1: An example of contrast sets the model picked up on. Taken from [43].

Diagnostic classifiers have since seen tremendous popularity, touted as a simple tool to pinpoint what information is stored in non-interpretable model internal representations. With regard to seq2seq generation tasks, however, they do not answer how well these representations translate to improved generations. To do so, one needs to also analyse the system's output, perturbing input in such a way that the desired property becomes clear.

A prominent analysis methodology that falls in that class are the contrast sets introduced by Burlot and Yvon [43], since repeated as part of the WMT'18 evaluation suites [44]. Contrast sets are built on the notion that an NMT system aware of morphology must be able to convert source-side interventions to similar alterations in the target-side output. By presenting the system with two sentences, one without and one with the intervened morphological feature, novel word-forms present in the intervened sentence indicate whether successful transfer occurred. For an example, see Fig. 3.1.

[43]: Burlot et al. (2017), 'Evaluating the morphological competence of Machine Translation Systems'

By no means is the use of contrast or challenge sets new [45], Burlot and Yvon include an overview or prior attempts, and clearly outlines differences. These distinguishing elements of analysis methods include,

1. holistic or analytic, general NMT score or specific to morphology
2. coarse or fine-grained, indicative of general difficulty or capable of identifying specific issues
3. hand-crafted or natural, does the input come from natural language or not
4. human judgement or automated, is a human judge required
5. the specific definition of the metric

At the time, they identify their work to be the only that was simultaneously analytic, fine-grained, and automated, all desirata. Prior work primarily relies on human judgement, a labour intensive process, or are limited to small sets of artificial sentence sets. However, they remain dependent on hand-crafted, artificial data. Likely for this reason, the contrast sets are limited to small, simple alterations that can be produced quickly. Neither situation allows for scaling to large-scale, system-wide comparisons.

Keeping the interventions simple in order to produce larger corpora is considerably more restrictive than it appears on the surface level. While the automated quality metric checks all altered words in the target-side output, the morphological assessment essentially becomes one-to-many relationship. Especially when translating into a morphologically richer language, the relationship between words in the source-side text to the target-side text is expected to be many-to-one or many-to-many;

sequences of function words in the source-side are typically compressed into a singular word in the target side. As such, important target-side morphological processes are invalidated if not present in the source-side language[1]. A secondary, related, concern, is whether or not they actually test for the system's morphological competence, and not just the transfer of morphology through the encoder-decoder bottleneck. The former is an altogether different concept than the latter. Especially with the perspective provided by Bisazza and Tump [42], the encoder's encoding of morphological information is not necessarily related to the decoder's output.

Interestingly, a concurrent work already addressed building contrast sets on target-side texts. Sennrich [46] instead deliberately introduce errors to ground-truth translations, with an error constrained to introducing disagreement belonging to a single morphological property[2]. Where Burlot and Yvon define their accuracy measure as one of the newly produced word-forms being marked for the intervened morphological feature, Sennrich define it as the model assigning higher probability to the ground-truth sentence than the altered, error-containing sentence. NMT systems score highly for all categories of introduced errors, although this is largely determined by the distance agreeing elements. While this does not require building hand-crafted test-sets, introducing errors to existing target-side data still does not scale easily, despite being analytic, fine-grained and otherwise fully automated.

Most recently, Pratapa et al. [47] design an NMT metric that automatically determines how well a sentence conforms grammatically with the ground-truth output. They achieve this by training a dependency parser, specifically augmented to be able to handle malformed text (like that produced by an imperfect translation system). The produced parsing provides per token a PoS, and across related tokens a dependency relation. This combination correspond directly to a grammatical rule, with agreement or presence of morphological markers indicating grammatical wellformedness. For `l'ambre`, they focus exclusively on agreement rules (e.g. adjectives modifying nouns should match in gender), and case or verb-form assignment (e.g. a pronoun that is the subject of a verb should be identifiable as nominative), although the exact form of these rules are extracted from the annotated treebanks used for the parsers. The corpus level score is computed as the mean of sentence-level scores.

Within Burlot and Yvon's framework, Pratapa et al. appear to meet all desirata. Their metric is specific to the grammaticallity of a system's output, it easily scales at inference time, and lends itself (with minimal adjustment) to an analysis of relevant rules. A further noteworthy attribute is independence from a reference translation. However, despite correlating reasonably well with human judgements of generated translations, when comparing to other metrics, it struggled identifying grammatically incorrect sentences from their corrected variants. Recent investigation [48] into the effect of domain-shift on automated dependency-parsers further bring into doubt the ability of pre-trained parsers correctly identifying grammatical rules present in malformed text.

1: For example, Dutch noun diminutives can yield valid translations from English despite remaining anonymous in the source-side text:

[EN] The horse ate the flower.
[NL] Het paard at de bloem.
[NL] Het paard at het bloempje.

Otherwise, casing in Czech is formed via noun declension, a relatively compact representation. To alter the casing in a language like English, function words need to be insertion and word order shuffling is likely required.

[46]: Sennrich (2017), 'How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs'

2: For example, the verb is made plural while the subject remains singular:

[Ref] ... dass der **Plan** verabschiedet **wird**
[Cntr] ... dass der **Plan** verabschiedet **werden**

Taken from [46].

[47]: Pratapa et al. (2021), 'Evaluating the Morphosyntactic Well-formedness of Generated Texts'

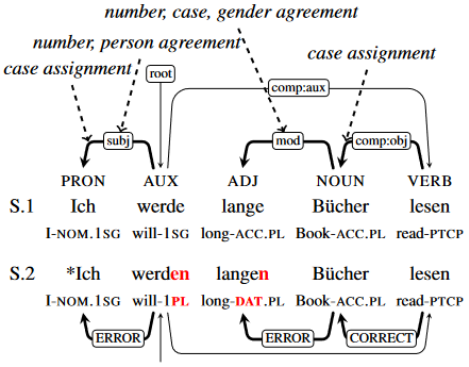


Figure 3.2: Parsed dependency relations of a generated sentence (bottom) compared to ground-truth (top). Each sentence would be evaluated separately. Taken from [47].

3.2 Conditional Generation of Morphologically Annotated Text

This section describes an analysis scheme that aims to determine the morphological competence of pre-trained NMT systems, in a similar fashion to the identified related works presented above. With regards to the aforementioned framework, it is analytic, as fine-grained as possible, uses natural data in a natural setting, and is fully automated. An important distinction is that it is metric agnostic, prescribing a method for collecting and sampling words from a parallel corpora, providing models with source- and target-side context just like training. The choice of metric, instead, determines what specific properties are tested for. Thus, evaluation occurs in vivo, aligning closely to the training algorithm used. Furthermore, unlike Pratapa et al., there is a minimal reliance on external resources, and those used are applied to natural language. Relative to other methodologies described, testing in this manner is limited only in the sense that available parallel corpora are limited, and perhaps most importantly, the analysis is not constrained to source-side morphological processes.

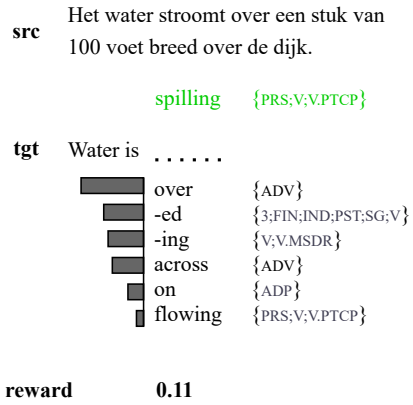


Figure 3.3: The proposed evaluation method for the morphological competence on NMT systems. Green gives the target word and its morphological tag set. The black words underneath the 'tgt' sentence the tokens actually produced, with the gray bars denoting their relative frequency. Reward in this instance is the expected IoU of the produced morph tags.

Consider some NMT system, parameterized by θ, that takes in some source-side text x, and produces a probability distribution over the next word, y_t, dependent on the previous target-side words, $y_{<t} = (y_1, \ldots, y_{t-1})$,

$$p(Y_t|x, y_{<t}) = f_t(x, y_{<t}; \theta). \tag{3.1}$$

To generate the next token, or more typically a sequence of sub-word units, one simply samples from said distribution and feeds the produced sub-word unit as additional context ($y_{<t+1} = (y_{<t}, \tilde{y}_t), \tilde{y}_t \sim p(Y_t|x, y_{<t})$), iterating until a complete word is produced. To accommodate common BPE/SP encoding schemes, in practise one produces units until a word boundary is detected, typically preprended to the next sub-word unit.

Let $R\left(\tilde{y}_t, y_t|x, y_{<t}\right)$ be some function that takes the produced sample and produces a gain/risk scalar (higher/lower values desired, respectively), indicating the expected proximity of the generation to ground-truth. Proximity, here, is a loosely defined concept, depending on the assumptions encoded in the underlying evaluation metric μ. If higher values are desired (e.g. accuracy, IoU) μ is a utility function, making R a gain function. Vice versa (e.g. Levenshtein distance), μ instead represents a loss, and R is referred to as a risk function. Regardless, it may be approximated as,

$$\begin{aligned} R\left(\tilde{y}_t, y_t|x, y_{<t}\right) &= \mathbb{E}_{\tilde{y}_t \sim p(y_t|x, y_{<t})}\left[\mu\left(\tilde{y}_t, y_t\right)\right] \\ &\approx \frac{1}{K}\sum_{k=1}^{K} \mu\left(\tilde{y}_t^{(k)}, y_t\right), \\ &\tilde{y}_t^{(k)} \sim p(Y_t|x, y_{<t}), \end{aligned} \tag{3.2}$$

with k denoting the k-th sample. From this, the expected risk for a task Task, defined as some distinct property of the ground-truth target-side word, y_t, such that $\mathcal{T}_i$ is the set of all y_t which posses property Task_i, may

be computed as,

$$\begin{aligned} R(\textsc{Task}_i) &= \mathbb{E}_{x,y_{<t}}\left[R\left(\tilde{y}_t, y_t|x, y_{<t}\right)|y_t \in \mathcal{T}_i\right] \\ &\approx \frac{1}{|\mathcal{T}_i|}\sum_{n=1}^{|\mathcal{T}_i|} R\left(\tilde{y}_t, y_t|x^{(n)}\right), \\ &\quad \left(x^{(n)}, y_{<t}^{(n)}, y_t^{(n)}\right) \sim \mathcal{D}_{\textsc{Task}_i}, \end{aligned} \tag{3.3}$$

with superscript n denoting the n-th datapoint in the annotated parallel corpus $\mathcal{D}$, indexed by $\textsc{Task}_i$. For evaluating the morphological competence of the word, a natural choice of Task is the word's morphological feature set, as described in Chapter 2.

Choices for $\mu\left(\tilde{y}_t^{(k)}, y_t\right)$

Using this definition of R, depending on the gain/risk μ used, the morphological competence of the NMT system f_θ can be evaluated, in the target-side output, at the level of individual words. Some viable choices, already used in Section 2.2.4:

1. the exact match between ground-truth and predicted word-form (utility)
2. the Levenshtein distance between ground-truth and predicted lemma (loss)
3. the intersection-over-union (IoU) between the ground-truth and predicted morphological tag sets (utility)

The first is directly accessible from the NMT system's output. The last two require re-tagging the entire sentence with only the target word, y_t, replaced with the generations $\tilde{y}_t$.

3.3 Effect of Morphological Features on Generating Czech Translations

The model chosen for evaluation come from Helsinki NLP's Opus-MT challenge [49]. Built on top of Marian-NMT [50], the models consist of a six layer encoder-decoder transformer architecture, with Sentence Piece (SP) [51] encoding as its sub-word tokenizer. All in all, the architecture most closely resembles `BART` [52] without layer normalization. By no means SoTA, the models do offer strong performance for its relatively low parameter count. The hyper-parameter set follows literature standards, and are made transparent to end users. Further helping its popularity is their inclusion in the popular HuggingFace `transformers` library [53] make these popular NMT systems for experimentation.

[49]: Tiedemann et al. (2020), 'OPUS-MT — Building open translation services for the World'

Spearheaded by Tiedemann, the OPUS project makes available a large number of parallel corpora, including thousands of language pairs across an equally diverse spread of domains [54][3]. Of special note are the Tatoeba challenge collections [55], making up the training data for the discussed model. This project aims to improve NMT performance for low resource languages, evaluating on the held-out Tatoeba corpora.

[54]: Tiedemann (2012), 'Parallel Data, Tools and Interfaces in OPUS'

3: For a global overview, along with performance indicators, see here.

[55]: Tiedemann (2020), 'The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT'

To allow for efficient estimation of $R(\textsc{Task}_i)$, balancing measurement error with tractability, a moderately sized subset is extracted. Specifically, the multi-domain training set of Ataman, Aziz, and Birch [7] is used, consisting of:

1. **Software**: Gnome, Tatoeba, KDE4
2. **Transcriptions**: Open Subtitles, TEDTalks
3. **News**: GlobalVoices
4. **EU**: EU bookshop

all taken from OPUS, with versions matching those presented by Ataman, Aziz, and Birch [7]. No learnable filtering is applied, however, with all corpora being mixed and then filtering out examples where the source-side and target-side texts fell outside of the 99th shortest percentile, and within 256 total tokens. The analysis is limited to Czech, although it is equally applicable to any of the other languages pairs for which this joint corpus can be constructed.

[7]: Ataman et al. (2019), 'A latent morphology model for open-vocabulary neural machine translation'

All available sentences are annotated with lemmas and morphological tag sets using the pre-trained `UDPipe2` model from Chapter 2[4]. An inverse index was constructed to allow rapid sampling and accessing of target words y_t and their context. Only one occurrence of $(\textsc{Tags}(y_t), \textsc{Lemma}(y_t))$, is kept per sentence, sampled uniformly from those present. Tag sets were capped at 1000 instances each, with stratified sampling applied over the lemmas, such that rare instances remain prevalent in the truncated dataset. On the other hand, tag sets with fewer than 32 samples were dropped entirely. In total 1,993 tag sets were recorded, yielding 790k instances over 930k sentences.

4: The best available at the time

For every $(x, y_{<t}, y_t)$ tuple, 24 samples were drawn from $p(y_t|x, y_{<t})$. Generation was allowed to terminate early, or at most at 5 tokens over the ground-truth length. To circumvent the noisy output introduced by label smoothing, top-P or nucleus sampling is used instead [56], truncating the distribution to the minimal set, i.e. the most probable tokens, to sum to P. A value of $P = 0.9$ is used throughout. Total processing time takes about 51 hours on a single NVIDIA TITAN RTX GPU.

3.3.1 Identifying Problematic Morphological Features

To showcase how these results might lead to a fine-grained signal as to which morphological features tend to induce errors, a model predicting expected risk from the gathered morphological features is estimated. Each word for which Eq. 3.2 has been estimated serves as a single data point, with the presence of a particular morphological feature (i.e. a dummy variable) serving as the independent variables, and $R\left(\tilde{y}_t, y_t|x, y_{<t}\right)$ as the independent variable. Emphasising interpretability, a linear regression model is estimated.

The morphological features used for the UD treebanks are structured into a hierarchy of non-overlapping categories. The presence of a tag in one category, or a combination of categories, indicate which other tags should be present also[5]. The top level features, common to all tokens, are the parts-of-speech. All other categories complete the paradigm for a particular lemma. For an example of how these features combine, see the dedicated site for Czech UD annotations or UniMorph's schema [12].

5: Czech nouns, for example, have an additional animacy dimension applied when marked for the masculine gender. When marked as feminine or neuter instead, this is not present. Envisioning this structure as a tree, the animacy dimension would be a child of the masculine gender, itself an instance of the gender dimension.

Table 3.1: Posterior of the dependent variables weights, resulting from a Bayesian linear regression for the IoU of the predicted and ground-truth morphological tag sets. The part-of-speech is entered as binary dependent variables, along side an global effect intercept. The model is drawn from an uninformative Beta(1, 1) distribution over each independent variable, whereas the parameter values are drawn from a Jeffreys-Zellner-Siow prior with an r-scale of 0.354. In total, at most 10k models are sampled using Bayesian adaptive sampling without replacement, with the presented posterior coefficients being model averaged. These hyperparameters largely reflect the default values used in JASP.

Category	Subcategory	Part-of-Speech	p(incl | data)	Mean	SD	95 CI LB	95 CI UB
Intercept			1.00	0.268	0.00	0.27	0.27
	Parts-of-Speech	Adjective	0.96	0.095	24.73	−0.01	0.16
		Participle (Adj)	0.89	−0.064	24.73	−0.17	0.00
		Adposition	0.47	0.012	24.73	−0.10	0.08
		Adverb	0.86	−0.056	24.73	−0.17	0.01
		Determiner	0.96	−0.026	24.73	−0.13	0.04
		Noun	0.96	0.107	24.73	0.00	0.17
		Numeral	0.96	0.028	24.73	−0.08	0.09
		Pronoun	0.96	−0.098	24.73	−0.21	−0.03
		Proper Noun	0.96	0.070	24.73	−0.04	0.14
		Verb	0.47	0.013	24.73	−0.09	0.08
		Participle (Verb)	0.93	−0.047	24.73	−0.15	0.02
Observations		510,778					
R^2		0.089					
$p(M^{(\text{null})}\|\text{Data})$		0.000					
$p(M^{(\text{best})}\|\text{Data})$		0.490					

Ideally, the analysis model incorporates this structure, but practically this proves difficult. As a compromise, interactions between the PoS with individual dimensions can also be considered. This effectively treats the PoS as the top layer, and all other dimensions as

The simplest model would be to simply find estimates for each part-of-speech in isolation,

$$R\left(\tilde{y}_t, y_t | x, y_{<t}\right) = \beta_0 + \sum_{\text{PoS}} \beta_{\text{PoS}} \mathbb{1}\left(\text{PoS}, \textsc{pos}\left(y_t\right)\right),$$

with β denoting the estimated coefficients, $\textsc{pos}\left(w_i\right)$ the parts-of-speech dimension taken from the full output of a morphological tagger, and $\mathbb{1}$ the indicator function ($\mathbb{1}(\cdot) = 1 \iff \text{PoS} = \textsc{pos}\left(y_t\right)$). The interpretation of the β_{PoS}s is simply 'if a token is marked for PoS, the expected risk shifts by β_{PoS} relative to the global average (intercept)'. All additional morphological features are conveniently averaged out.

Table 3.1 presents such a model, with the morphological tag set IoU as the dependent variable, estimated using JASP [57][6]. No attempt at parsimony is made, although the discussion naturally limits itself to those for which $p(\text{incl}|\text{data}) > 0.95$. The coefficients presented come from the model averaged posterior. Differences between parts-of-speech already become apparent. Clear winners appear to be nouns, adjectives (exclusively those that modify nouns) and proper nouns. Contrasted to these are the pronouns, with generated words sharing only 17% of the morph tags of the ground truth word. Pronouns typically replace nouns and proper nouns in sentence, potentially indicating co-reference resolution is problematic for the NMT system. Determiners, carrying the same function as pronouns but for adjectives, also see a negative

6: The output provided is as given by JASP. While the standard deviations appear suspect, the confidence intervals do differ indicating this is likely just a misprint.

coefficient, although in magnitude nowhere close to the pronouns.

Regardless, the analysis is hampered by high standard deviations, with many of the independent variables including 0 in their confidence intervals. A possible reason is an underspecified model: particular features contain more information regarding risk/gain propensity than the part-of-speech alone is able to provide. For example, infinitive verbs should be simple, typically in lemma form, whereas ones inflected for tense might be troublesome. To that end, the analysis is expanded, including for each parts-of-speech, the relevant morphological features. The estimated effects indicate the degree to which a feature being present in a word, shifts the expected risk from the average for that word's part-of-speech. In essence, this adds the second layer to the regressions,

$$R\left(\tilde{y}_t, y_t | x, y_{<t}\right) = \beta_0 + \sum_{\text{PoS}} \mathbb{1}_{\text{PoS}}(\text{PoS}(y_t)) \left(\beta_{\text{PoS}} + \sum_{\text{tag}} \beta_{\text{PoS, tag}} \mathbb{1}(\text{tag} \in \textsc{Tags}(y_t)) \right),$$

with $\beta_{\text{PoS, tag}}$ being the effect of the PoS being marked for tag. With the many available dimensions, and the many individual tags within each, the produced model contains a large number of coefficients. For brevity's sake, these are presented in Section B.1. Differences within a dimension can be seen clearly, for example when considering noun casing. Again, adjectives and determiners score lowest overall.

3.3.2 Identifying Common Confusion

Using the gathered dataset, this subsection provides another possible qualitative analysis technique. It is meant to display how expressive data is, and make clear how such data might be used for building a task scheduler for morphologically aware training (a use-case critical to Chapter 4). The intent of this analysis is not to provide rigorous evidence as to problematic morphological features, but instead to diagnose holistically which generations replace which. Thus, this subsection deviates somewhat from the prescribed method of Section 3.2, but the changes are easily made.

Specifically, the only necessary change is recording the generated morphological tag sets instead of a risk/gain function. These are grouped by the ground truth tagsets, essentially providing information as to which inflections are generated when presented with a particular morphological feature set.

First, a joint distribution $p(\textsc{Tags}(y_t), \textsc{Tags}(\tilde{y}_t))$ between all morphological tag sets known to share a lemma[7], is estimated on combinations between the ground truth and predicted morphological feature sets. This distribution is in terms of tag sets, not individual tags. To convert to one encoding the confusion probability between individuals tags, this needs to be converted to a marginal conditional distribution, i.e. $p(\text{tag}_a \in \textsc{Tags}(y_t) | \text{tag}_b \in \textsc{Tags}(\hat{y}_t), \text{tag}_b \in \text{mistakes})$. This may be read as the probability that tag_a is in the tag set of the ground-truth word-form, given that tag_b is mistakenly present in the tag set of the generation. Otherwise, given the mistake is known to be tag_b, what is the probability it should have been tag_a. Marginalization is complicated somewhat due to the definition of 'mistake' with respect to set prediction:

7: As determined by the annotated parallel corpus. This does limit the instances somewhat, but given the evaluation dataset was present during NMT pre-training, this ensures the model is aware of those lemmas in those word-forms.

Figure 3.4: The marginal confusion matrices, indicating the error type between the predicted (columns) and ground truth (rows) tags. Each cell indicates a certain conditional probability. Bright colours indicate high prevalence. Cells are blocked into their respective category, with the first (top left) being the parts-of-speech. The individual tags are given by the bottom matrix's column headers, colour coordinated with their respective category. Vertical text indicates which type of mistake is being considered.

1. **False positive**, the system produced a word marked for a tag *not present* in the ground truth tagset

$$p(\text{tag}_a \in \textsc{Tags}(y_t) | \text{tag}_b \in \textsc{Tags}(\tilde{y}_t), \text{tag}_b \notin \textsc{Tags}(y_t))$$

   Here, the mistake is attributed uniformly to all true positive tags $\text{tag}_a \in \textsc{Tags}(y_t)$
2. **False negative**, the system produced a word *not marked* for a tag present in the ground truth tagset

$$p(\text{tag}_a \in \textsc{Tags}(y_t) | \text{tag}_b \notin \textsc{Tags}(\tilde{y}_t), \text{tag}_b \in \textsc{Tags}(y_t))$$

   Here, the mistake is attributed uniformly to all predicted positive tags $\text{tag}_b \in \textsc{Tags}(\tilde{y}_t)$
3. **Substitution**, the system produced a word marked for a tag that was either a false positive or a false negative. Otherwise, one may read this as, 'given what was erroneously produced, I can swap it with some forgotten tag'. Unlike the previous two, the mistake is attributed only to other mistakes, invalidating cases where either no false positive or false negative occurred.

Precisely such marginal confusion matrices are presented in Fig. 3.4. Obviously, the majority of substitutions occur with the morphological category. Within the parts-of-speech, nouns and adjectives are commonly substituted, although adjectives also get swapped with determiners. Generally, looking now at the false positives, the NMT system appears to be too keen in producing nouns and adjectives, with conjunctions being especially neglected in favour of nouns.

With regards to casing, pronouns seem to be infrequently marked for case, especially when instrumental. Likely, this implies generated pronouns typically do not agree in case inflection with the nouns they replace. Interestingly, this pattern does not appear when considering dimensions like gender & animacy or number, other markers applied to nominal words.

The number, comparison and polarity dimensions are most often confounded, both within the category (see blocks in 'Substitutions'), and outside of it (see the row blocks in 'False Positives' or 'False Negatives'). Determiners should be marked as singular more often, whereas numerals are not marked plural often enough. The high confusion between comparative and relative tags indicates trouble separating relative and superlatives adjectives and adverbs. Specifically, the NMT system produces more superlatives than relatives, when the latter suffices. Most egregious of all three, though, is the polarity dimension. Mostly independent of all other dimensions, words tend to marked positive (i.e. not negative) far more often than should be. Negation, and/or agreement of negation, is thus clearly a prevalent issue.

3.4 Discussion

As promised, this chapter covers existing methods for assessing the morphological competence of neural language models, and designs a new technique that addresses some existing pitfalls. This then lends itself

to handy auxiliary analysis methods which can identify morphological features, in the target language. In the grander scope of this thesis, however, these are merely secondary results. The main reason for this error methodology is be able to design a task sampler and an assessment tool to detect improvement in morphological competence post-adaptation.

One flaw inherent to the proposed methodology is the restriction of only considering the next generated word. Effectively, this also measures how well the NMT system adheres to the word-order prescribed by the target-language. It could very well be that the properly inflected word-form is generated later on in the sentence. The contrasts set, which simply look at the difference pre- and post-perturbation, do not suffer from this. Another issue, applicable when re-tagging the sentence with the sampled word-forms, is similar to that of `l'ambre`. The model is forced to infer using malformed input, or generally input not present in its training domain. In this case, the model might defer its decision to contextual information, which are left unaltered, and could inflate scores unfairly. The effect on risk estimation is clear, with the estimated regression models achieving low explained variance (R^2), and the finding that only 36% of instances see the ground-truth lemma produced at all.

How to deal with either of these issues within the proposed framework is not immediately clear. One could draw inspiration from Burlot and Yvon [43], and allow the NMT system to generate the entire remaining sentence and explicitly searching for the ground-truth word-form or morphological feature set (the latter allowing synonymy). This comes at a drastically increased expense, however, which could make evaluation practically intractable. Another possibility would be to use Sennrich's [46] method instead, relaxing the expected risk estimation. Instead, the model is allowed to look at the generated sentences $(y_{<t}, \tilde{y}_t, y_{>t}$, and based on the perplexity it assigns, a re-ranking could be performed. Taking the example provided in Figure 3.3, while the synonym 'flowing' is produced relatively infrequently, when swapped in place of 'spilling', the NMT system might naturally prefer it over the replacement 'over'. While this does not assess $p(Y_t|x, y_{<t})$, it does test, to some extent, how well the system is capable of detecting the correct inflection, regardless of how well it generates it.

Adapting NMT Systems for Morphological Awareness

4

This final content chapter finally sets out to do what has been hinted at earlier: teaching pre-trained NMT systems to morphologically inflect. As stated in the introduction, the trick here, is to retain the architectures and performance of existing NMT systems. Morphological awareness has been injected into every phase of the typical NMT system; from pre-processing, to decoding. The related works section covers a representative sample of these works, with particular emphasis on a recurring hypothesized learning mechanism: 'copy-and-inflect'. The next section takes a step back and covers a learning paradigm no yet applied, namely gradient-base meta-learning. While the connection to morphological awareness is not immediately clear, the following chapter covers an episodic learning framework that, through only altering the sampling method, should target the 'copy-and-inflect' mechanism directly. This is operationalised in a series of experiments, with evaluation of the adapted systems occurring from the viewpoint of a general NMT system, and a dedicated morphological inflection module. While improvements in both categories can be seen, these tend to occur separately from each other. This indicates that despite success in teaching morphological inflection, the devised learning framework does not necessarily align with improved translation capacity.

4.1 Related Work

A plethora of approaches inducing morphological awareness in NMT systems already exist. These approaches have been applied to all facets of the standard NMT pipeline; from sub-word tokenization methods using morpheme boundaries to constrained decoding. Broadly, existing methods fall into three categories, i) informed tokenization, ii) modelling architectures and iii) data/objective augmentation. The first two categories necessitate architectural changes, or at least significant re-training of existing architectures, invalidating them from use for black-box systems. The last suffers from the same issues, but can often be recast as supplemental to pre-trained models. As such, the discussion will focus primarily on Data & Objective Augmentation

4.1.1 Informed Tokenizers & Architectures

With respect to modern seq2seq architectures, the input embeddings and output classification are typically considered the slowest and most resource intense operations throughout. Especially for outputting sequences with large vocabularies, with morphologically rich languages being extreme outliers, the parameter count required to perform word-level classification would dwarf the parameters used for general language understanding. Using characters instead, limiting output symbols to small set while retaining infinite productivity, incurs far longer sequences,

which increase latency, memory consumed and increase modelling difficulty. In order to strike an efficient balance between the two, sub-word tokenization is a commonly applied pre-processing step.

As mentioned in earlier chapters, SoTA models rely primarily on bytepiece or the related SentencePiece encodings [51, 58]. These systems retain productivity, with models theoretically capable of producing rare word-forms or new word-forms of existing lemmas, while striking a data-driven balance between character and word-level encoding of language. They are not, however, morphologically informed.

From Mielke et al.'s [59] review, it becomes clear that sub-word tokenization is by no-means a new idea, with a long and varied history. More importantly, they highlight the difficulty in beating BPE baselines, with limited results when using unsupervised morphological segmentation methods, and supervised methods showing benefits only for lower-resource languages with particular word-formation processes.

A successful line of research, bridging both the use of morphologically informed tokenizers and models with inductive biases for morphological inflection, is that of Dugyu Ataman and co-authors. Starting in 2017, Ataman et al. [61] start by training an unsupervised hidden Markov model variant that balances vocabulary reduction with segmentation. They present moderate increases to general NMT metrics over using just BPE based approaches. By 2018, this is followed up by removing the pre-processing altogether, and determining a character-word trade-off at the model level. Ataman and Federico [62] circumvent the standard BPE/word embedding layer, and instead train a character n-gram `RNN` with a time-pooling operation to generate from characters, a dynamic word representation. In effect, this is similar to the `char2vec` modules used in Chapter 2. The use of composition again yields moderate gains in NMT metrics, further improving upon the use of an unsupervised morphological tokenizer as a pre-processor. Ataman et al. [60] take this notion of composition one step further, not just encoding language at the character level prior to contextualizing, but also decoding at the character level again. This hybridizes character-level NMT. While the manage to show only small gains in NMT scores, using the contrast sets [44] described in Chapter 3, they manage to show improved morphological competence, with the gain being especially prominent for more morphologically complex languages like Arabic and Turkish.

Research into character-based NMT remains active. With the advent of self-attention based architectures, whose performance scales quadratically with sequence length, incorporating character level information while retaining the reduced latency relative to `RNN`s remains an open issue. Recent advances, like `CANINE`, still cannot perform at the level of NMT quality as standard sub-word tokenizer pre-processed models do, and contrary to expectations, are not more morphologically competent [**ibovicky-etal-2022-dont**].

Furthermore, while such models operate on more morphologically inclined sub-word units, there still is no explicit inductive bias for morphological inflection. Even the hierarchical models still largely contextualize representations at the word-level, with no guarantee that morphological information is considered. Ataman, Aziz, and Birch [7] enforce a model reminiscent of how humans generate words from morphological

[59]: Mielke et al. (2021), 'Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP'

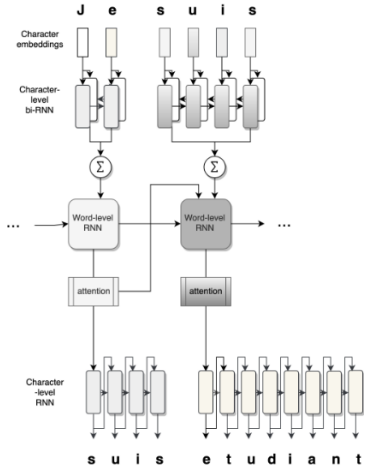


Figure 4.1: Hierarchical representations of characters and words should give the best of worlds. Taken from [60].

[61]: Ataman et al. (2017), 'Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English.'

[62]: Ataman et al. (2018), 'Compositional Representation of Morphologically-Rich Input for Neural Machine Translation'

[60]: Ataman et al. (2019), 'On the Importance of Word Boundaries in Character-level Neural Machine Translation'

Features	Output	English Translation
[1,1,1,1,1,1,1,1,1,1]	git	go *(informal)*
[0,1,1,1,1,1,1,1,1,1]	'*a* git	*to* go
[0,1,0,1,1,1,1,1,1,1]	'*da* git	*at* go
[0,0,0,1,1,0,0,1,1,0]	gidin	go *(formal)*
[1,1,0,0,0,0,1,0,1,1]	gitmek	to go *(infinitive)*
[0,0,1,0,0,0,0,0,0,1]	gidiyor	(he/she/it is) going
[0,0,0,0,0,0,0,0,1,0]	gidip	*by* going *(gerund)*
[0,0,1,1,0,0,1,0,1,0]	gidiyoruz	(we are) going

Figure 4.2: Perturbing the morphological feature vector yields different inflections of the same lemma. Taken from [7].

[7]: Ataman et al. (2019), 'A latent morphology model for open-vocabulary neural machine translation'

information, decomposing the hidden representations into a semantic (i.e. the lemma) and morphological feature set. By treating this task as latent variable inference, this process can happen unsupervised, makeing the model essentially a `VAE` [63]. The encoder consists of the source-side NMT encoder and the target-side character to word-level model described earlier [60]. The latent space is built from a multidimensional Gaussian semantic and a multidimensional Beta-esque morphological inflection distribution. The decoder consists of an attention mechanism over the encoder's hidden states, and a character-level decoder. Not only do general NMT metrics see a modest improvement, there is evidence that it generalizes slightly better to unseen word-forms. When analyzing samples from the discrete morphological latent space, perturbations show it determines which affixes are generated, while the lemma is left unchanged.

4.1.2 Data/Objective Augmentation

Unlike traditional data augmentation techniques, which seek to affordably extend or regularize the training data with unseen but plausible examples [64], when proposed for morphologically aware NMT, the main purpose is providing the model with an additional form of supervision. This supervision can occur at either the encoder or decoder, but importantly, always seeks to make explicit the morphological features present in the target-side text. When applied to the decoder, or target-side, it is typically the labels that are altered. In essence, the NMT system is forced to produce representations useful both for seq2seq text generation and predicting morphological features. Augmentation of the encoder input is a more recent research avenue, primarily drawing inspiration from lexically constrained decoding [65, 66]. The expected benefit of these techniques is the capacity to disentangle a word-form's lemma from affixed morphological features.

Target-side Augmentation

While not directly related, the approach tailored by Nadejde et al. [67] has proven influential. They note the necessity of encoder-decoder models to learn target-side syntax, and aim to directly improve the decoder's ability to do so by interleaving syntax information in the target-side sequence. The NMT system is now required to output both, iterating a word with a syntactically motivated tag. Interestingly, performance gains only held when sharing both the encoder and decoder blocks.

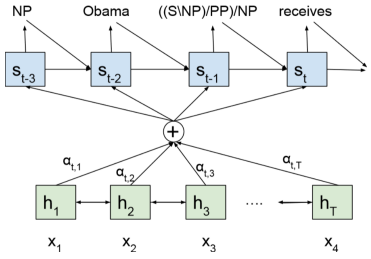


Figure 4.3: CCG supertag interleaving in the target-side text. Taken from [67].

Directly inspired by these experiments, but focused on morphology, are the works of Tamchyna, Marco, and Fraser [68], Conforti, Huck, and Fraser [69] and Marco, Huck, and Fraser [70]. Where Nadejde et al. [67] [67] require the decoder to produce the same information twice, Tamchyna, Marco, and Fraser [68] instead require the system to disentangle stem from affix, alternating between producing the lemma and the morphological feature set. Finally, the combination of the two are deterministically mapped to their surface form. The authors postulate that this requires improved generalizability, mapping words to lemmas and vice versa, a claim repeated by later work. By limiting the decoder's textual output to lemmas and morphological features results in a 1.7

[67]: Nadejde et al. (2017), 'Predicting target language CCG supertags improves neural machine translation'

[68]: Tamchyna et al. (2017), 'Modeling target-side inflection in neural machine translation'

[69]: Conforti et al. (2018), 'Neural morphological tagging of lemma sequences for machine translation'

point BLEU English-Czech gain. In a follow-up paper, Conforti, Huck, and Fraser [69] takes the two-step NMT approach to an extreme, requiring of the decoder only lemmas as output, and using a secondary neural model to predict the eventual word forms. While the NMT system produced coherent text, and the correct lemma slightly might often, the word forms tended to confuse morphological features more often; decoupling lexical choice from word formation entirely, leads to poorer NMT systems for morphologically rich languages.

Recently, Marco, Huck, and Fraser [70] compare and contrast a variety of methods in a similar vein, demarcating methods as knowledge poor (little linguistic information is provided to the NMT system) or knowledge rich (much formal linguistic information is provided). The former category includes linguistically motivated segmentation tokenizers, whereas the latter includes Tamchyna, Marco, and Fraser's two-step lemma + morphological tag set approach. Focusing on English to German, they show knowledge rich approaches consistently outperform both baselines and knowledge-poor approaches on small to medium-sized datasets. This effect was most prominent when considering out-of-domain test corpora and unseen word-forms.

Dalvi et al. [71] independently experiment with variations to interleaving, defining successively looser joining of the objectives on top of the final hidden representation layer. Specifically, considered are i) right-side appending of the morphological feature set sequence, ii) generating either word-form or feature set sequences through the same decoder, essentially treating each as a separate target-side language, and finally iii) multitask learning, with separate classification heads on top of the decoder body working in parallel. Likely due to taxing the long-range dependencies of the recurrent architectures, only the latter two techniques proved successful. When tuning a loss balancing parameter, especially multitask learning yielded significant performance gains across a variety of language directions.

Planetenbewegungen ('planetary motion')

```
Planet<NN>bewegen<V>ung<SUFF><+NN><Fem><Gen><Pl>
```

$planet_{NN}$ $move_{V}$ $ment_{SUFF}$

Figure 4.4: Annotated lemma output for a compound noun in English-German translation. Taken from [70].

[70]: Marco et al. (2022), 'Modeling Target-Side Morphology in Neural Machine Translation: A Comparison of Strategies'

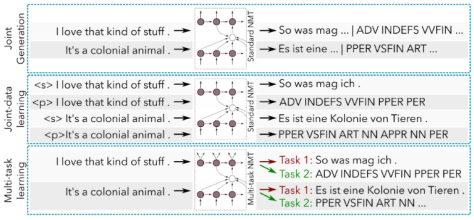


Figure 4.5: Various multitask learning objectives, joined at different levels. Taken from [71].

[71]: Dalvi et al. (2017), 'Understanding and improving morphological learning in the neural machine translation decoder'

4.1.3 Source-side Augmentation

Term injection techniques, when presented as a training regimen, require of the NMT system some word form to be present in the target-side generation [72]. This can be achieved by appending the target-side constraint (as a word form) to the source-side text, an annotation scheme dubbed Exact Term Annotation (ETA). At inference time, the surface forms of the lexical constraints must therefore already be known.

Beyond whether this assumption is feasible, for successful constrained decoding, the decoder merely has to *copy* the target side word form. Hence, models trained with term-injection in mind will likely prove incapable of generalizing to all surface-forms of a lemma's paradigm. For morphologically rich languages, with their trademark complex yet sparsely evidenced word-formation processes, this presents a significant problem.

Both Exel et al. [73] and Bergmanis and Pinnis [74] instead argue for softer constraints; the NMT system must remain free to infer the present linguistic phenomena and its impact on the surface form of the injected lexical item. In order to achieve this, they inject into the source-side text

[72]: Dinu et al. (2019), 'Training Neural Machine Translation to Apply Terminology Constraints'

[73]: Exel et al. (2020), 'Terminology-Constrained Neural Machine Translation at SAP'

[74]: Bergmanis et al. (2021), 'Facilitating terminology translation with target lemma annotations'

the lemmas of the requested word, dubbed Target Lemma Annotation (TLA). The model is thus trained to *copy-and-inflect* the injected constraint into the target-side text. Effectively, both encoder and decoder are taught to disentangle a lemma from its surface form.

Initially presented by Exel et al. [73], Bergmanis and Pinnis [74] apply this setup to lemmas far removed (in terms of Levenshtein distance) from their word forms, and rigorously test it by including morphologically rich target-side languages to translate into. Compared to ETA, they achieve higher `BLEU` scores (0.3-7.5 points gain), despite meeting the constraint less often. The largest `BLEU` gains were for the more morphologically complex languages. Perhaps more impressively is the reported improvement in the quality of the constrained word-form. Not only does lemma injection produce the correct lemma more often, the NMT system is significantly better at inflecting it properly compared to used baselines. Furthermore, of word forms produced *not* present in the training vocabulary, 62.5% were correctly inflected. They conclude that the model's ability to morphologically inflect is productive.

Extending the above presented methodology to Czech, Jon et al. [75] show that for such a morphologically complex language, TLA or the *copy-and-inflect* mechanism is crucial for properly inflected target-side constraints. They train using words lemmatized via `UDPipe2`, and note that pre-trained transformers can be effectively fine-tuned for term injection.

All presented TLA systems show reduced constraint coverage; the NMT system is less likely to produce the desired word form, regardless of morphological features. Xu and Carpuat [76] attempt to counteract this by training or defining a secondary module that inflects target-side lemmas using source-side text as conditioning information. The produced word-form can then be incorporated as a constraint as ETA. Once again, incorporation of morphologically motivated constraints improved the model's accuracy when inflecting. In fact, when comparing to other terminology constraint methods, while the lemma is often correctly incorporated in the target-side sentence, when inflection is required, the surface form typically is not. Compared to TLA, using a separate inflection module allows the NMT model to significantly boost term-accuracy when confronted with rare or unseen word-forms, better handling sparsity inherent to morphologically complex languages. Otherwise, little difference was found when training with *copy-and-inflect* in mind.

[75]: Jon et al. (2021), 'End-to-End Lexically Constrained Machine Translation for Morphologically Rich Languages'

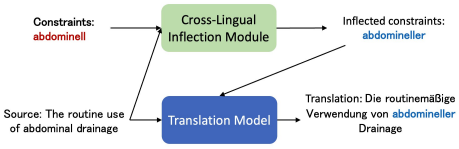


Figure 4.6: From TLA to ETA via a secondary rule-based module.

[76]: Xu et al. (2021), 'Rule-based Morphological Inflection Improves Neural Terminology Translation'

4.2 Gradient-based Meta-learning

This section introduces the motivation and notation common to gradient-based meta-learning (GBML) frameworks. The differences relative to a standard machine learning setup are highlighted. It should be noted that the use of GBML here differs from those typically found in the literature, namely inductive bias learning (or, to frame it in terms of few-shot learning, zero-shot). An overview of why certain GBML techniques can and cannot achieve this, is briefly discussed here, and in far more detail in Section C.1.2.

Model-agnostic meta-learning (MAML), as introduced by Finn, Abbeel, and Levine [77], has proven to be a seminal learning paradigm. Using bi-level optimization, strong initialization and fast adaption to a wide array of tasks can be directly learned simultaneously, irrespective of the specific architecture. In the previous sentence, reference is made to two crucial properties of meta-learning. 'Strong initialization' is encapsulated in the meta-model, an initial parameter set shared across tasks, which determines the performance of the task prior to seeing any data. 'Fast adaptation', instead, is the ability of the learner to rapidly incorporate new experience, yielding a task-specific parameter set typically denoted the task- or episode-model.

[77]: Finn et al. (2017), 'Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks'

The notion of tasks is left purposefully vague, although somewhat similar to common machine learning tasks. In developing GBML systems, tasks are commonly defined as subsets of all possible classes present in the dataset. A motivating example used by Finn, Abbeel, and Levine was defining tasks as regressing data from differing, but related, sinusoidal functions. In NMT systems, a common task definition are different languages [78, 79], leveraging high-resource languages to improve low-resource language translation.

The overall learning objective can be succinctly captured as,

$$\min_{\theta^{(\text{meta})}} \mathbb{E}\left[\mathcal{L}\left(f_{\theta^{(\text{episode})}}\left(D_{\text{Q}}\right); D_{\text{S}}\right)\right]. \tag{4.1}$$

In effect, this loss emulates the generalization capacity of the task model, using the query data (D_{Q}) as an unseen validation set, while the support set (D_{S}) simulates the training data. While the support and query sets might share the same underlying dataset, the support and query sets are assumed to be non-overlapping, such that the query set remains unseen during adaptation $\left(\mathcal{D}^{\text{S}} \cap \mathcal{D}^{\text{Q}} = \varnothing\right)$. The loss is ultimately w.r.t. $\theta^{(\text{meta})}$.

Unlike standard machine learning setups, MAML takes an episodic training scheme. A single iteration (optimizer step), consists of a number of episodes, i.e. a meta-batch. Each episode considers a single task, sampled from some task-distribution, $\textsc{Task}_i \sim p(\textsc{Task})$. In vanilla MAML, the task-distribution is uniform, although any sampling method could be employed. At the start of the episode, the meta-model's weights are copied as an initialization point for the selected task, $\theta_0^{(\text{episode})} \leftarrow \theta^{(\text{meta})}$. The episode model is then allowed to adapt to the task via K steps of stochastic gradient descent (SGD):

$$\theta_{k+1}^{(\text{episode})} \leftarrow \theta_k^{(\text{episode})} - \alpha \nabla_{\theta_k^{(\text{episode})}} \mathcal{L}_{\textsc{Task}_i}\left(\mathbf{y}^{\text{S}}_{\textsc{Task}_i}, f_{\theta_k^{(\text{episode})}}\left(\mathbf{x}^{\text{S}}_{\textsc{Task}_i}\right)\right). \tag{4.2}$$

The SGD update step corresponding to Eq. 4.1 then becomes,

$$\theta^{(\text{meta})} \leftarrow \theta^{(\text{meta})} - \beta \nabla_{\theta^{(\text{meta})}} \sum_{n_{\text{episodes}}} \mathcal{L}_{\textsc{Task}_i}\left(\mathbf{y}^{\text{Q}}_{\textsc{Task}_i}, f_{\theta_K^{(\text{episode})}}\left(\mathbf{x}^{\text{Q}}_{\textsc{Task}_i}\right)\right). \tag{4.3}$$

In principle, this should encourage $\theta^{(\text{meta})}$ to move towards a point from which adaption to tasks within $p(\textsc{Task})$ is quickly achieved. This can be done by either learning a parameter set proximal to optimal for all tasks, or one which rapidly incorporates novel information without overfitting. Ideally, of course, both are achieved.

While not readily obvious, it is important to notice that Eq. 4.3 incurs a second-order gradient. Here $f_{\theta^{(\text{episode})}}$ is defined as an updated version of $f_{\theta^{(\text{meta})}}$, such that finding the update direction requires computing a gradient through the unrolled updates;

$$\nabla_{\theta^{(\text{meta})}} \left(\theta^{(\text{meta})} - \sum_{k=1}^{K} \nabla_{\theta^{(\text{episode})}_k} \mathcal{L}_{\text{Task}_i} \left(\mathbf{y}^{\text{S}}_{\text{Task}_i}, f_{\theta_k^{(\text{episode})}} \left(\mathbf{x}^{\text{S}}_{\text{Task}_i} \right) \right) \right).$$

This can be prohibitively expensive, especially for the larger models that dominate the current NLP landscape. To circumvent this, first-order approximations can be made. Simply dropping higher-order terms leads to first-order MAML (foMAML), a relatively simple approximation method, which can be even further simplified [80]. Fallah, Mokhtari, and Ozdaglar [81] show theoretical convergence for MAML, but cannot do so for foMAML. For most applications, these approximations perform well enough to warrant ignoring higher-order terms. First-order updates do, however, work as an effective pre-training technique, and can help stabilize the performance of MAML trained few-shot classifiers [82]. A full GBML pseudo-code implementation can be found in Section C.1.1.

[81]: Fallah et al. (2019), 'On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms'

[82]: Antoniou et al. (2018), 'How to train your MAML'

While this section has motivated MAML as a gradient-based meta-learning framework, future sections will make use of the related Almost No Inner Loop (ANIL) algorithm instead. Introduced by Raghu et al. [83], this algorithm allows only the classification head to adapt to the support set, such that the meta-model's weights prioritize strong cross-task initialization. Given f_θ represents a pre-trained NMT system, and should stay an NMT system with a slightly altered inductive bias, this was deemed preferable to weights that quickly adapt; the two objectives are related, but not necessarily overlapping. This was consistent with early experiments, with diverging support loss but low query loss being quickly achieved without ANIL. An extended discussion, with a discussion of empirical evidence provided by the experiments conducted, can be found in Section C.1.2.

[83]: Raghu et al. (2020), 'Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML'

4.3 Learning Copy-and-Inflect via Morphological Cross Transfer

This section introduces a novel interpretation of gradient-based meta-learning approaches, aimed specifically at enforcing an inductive bias for morphological inflection in pre-trained NMT systems. The discussed learning algorithm is left intact, with alterations made only to the task scheduler, $p(\text{Task})$, and the method through which the support and query sets, $(\mathcal{D}^{\text{S}}, \mathcal{D}^{\text{Q}})$, are drawn.

Recent work in source-side data augmentation has indicated that allowing models to 'copy-and-inflect' lemmas to word-forms in the target-side language is beneficial in teaching systems about inflectional morphology. Rather than stochastically appending information on the encoder side, instead the lemma to word-form process occurs in both the support and query sets, disjointly. The support and query sets each contain one morphological feature set. Lemmas present in one morphological feature set in the support set, are present in the other morphological feature in

the query set, and vice versa. Samples sentences are certain to contain the lemmas, and are split on possessing the first tag set in the support or query sets. During adaptation, the model is allowed to learn the word formation process, but for successful generalization, it must be able to disentangle the lemma from the word-form representation of the word. In effect, this provides the NMT system with the same information as the posited 'copy-and-inflect' mechanism, except it is now implicit, and does not require adjusting the encoder to pass on target-side lexical information.

As suggested earlier, a natural task definition for this scheme is the combination of a morphological feature set and a lemma edit script. Their combination specifies a set of related words, possessing the same morphological markers, with the included lemmas containing the intended lexical meaning. While such information is typically not present in a standard parallel corpus, the morphological taggers trained in Chapter 2 can easily annotate arbitrary datasets with high fidelity. From the annotated sentences, and inverted index can be easily constructed, providing per task the sentence and token locations of relevant word-forms. The process is graphically depicted in Fig. 4.7, and a specific example is provided in Appendix C Figure C.1.

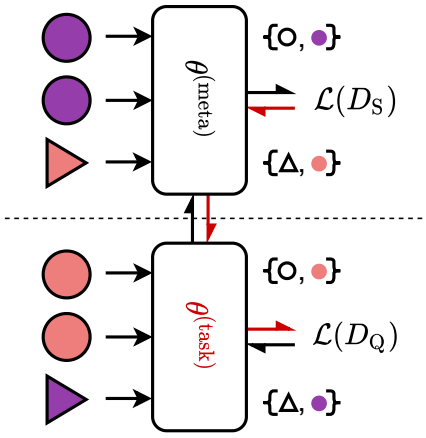


Figure 4.7: Cross-transfer of properties in an episodic learning framework. Two entangled properties (here, colour and shape) are presented in the support set. For successful generalization to the query set, the model must learn to disentangle the properties, and recombine them during output.

It is important to note that from the model's perspective, nothing has changed from its training regime. The only tangible difference is that loss computation directly assesses the model capacity to distinguish between lemmata and affixes. However, while meta-learning is commonly applied specifically to enable few-shot learning capacity, here it is only provided as a means to an end. Here it is merely an additional fine-tuning phase, and rather than learning-to-learn from limited examples, it need simply learn the concept being taught before returning to standard NMT inference.

Not all morphological features are equal, in the eyes of an NMT system, as Chapter 3 has made clear. From the discussion on MAML and its derivatives, a task distribution can be easily incorporated into a GBML framework. Despite this, research into useful distributions is not forthcoming. Existing work relies primarily on the abstract concept of task difficulty, typically found via some external tool, but seems divided as to whether a non-uniform distribution produces improves generalization [84, 85]. In the general N-way K-shot classification case, generating an joint distribution over all tasks is practically intractable without limiting oneself to a distribution of $N = 2$ [86]. With morphological cross-transfer in place, however, the number of tasks is naturally set to 2. Furthermore, precisely such a joint distribution over tasks has already been estimated for visual analysis in Figure 3.4. These record example commonly confused for one another, with the NMT system producing word-forms with erroneous morphological markers when presented with a particular feature in the target-side input.

In combination, an annotated parallel dataset and a distribution over difficult annotations that need disambiguating, should provide a (hopefully) powerful tool for teaching pre-trained architectures to disentangle lemma from affix tokens; teaching an inductive bias towards morphological inflection similar to the 'copy-and-inflect' mechanism. Algorithm 1 provides a pseudo-code implementation of morphological cross-transfer.

Algorithm 1 Cross-transfer

Require: A parallel dataset $\mathcal{D}$ indexed by tasks, $\textsc{Task}_i$, consisting of source-side text x, target-side text y; a joint distribution over tasks $p(\textsc{Task}_i, \textsc{Task}_j)$

function CrossTransfer($p(\textsc{Task}_i, \textsc{Task}_j)$, $\mathcal{D}$)
  Sample $\textsc{Task}_1, \textsc{Task}_2 \sim p(\textsc{Task}_i, \textsc{Task}_j)$
  Get (a subset of) all common lemmas,

$\overline{\text{lemmas}} = \{\textsc{Lemmas}(y) | \forall y \in \mathcal{D}(\textsc{Task}_1)\} \cap \{\textsc{Lemmas}(y) | \forall y \in \mathcal{D}(\textsc{Task}_2)\}$

  Split common lemmas into two sets $\overline{\text{lemmas}}_\text{A}, \overline{\text{lemmas}}_\text{B}$

  From $\mathcal{D}(\textsc{Task}_1)$, place examples with lemmas
    $\overline{\text{lemmas}}_\text{A}$ in D_S, and
    $\overline{\text{lemmas}}_\text{B}$ in D_Q

  From $\mathcal{D}(\textsc{Task}_2)$, place examples with lemmas
    $\overline{\text{lemmas}}_\text{B}$ in D_S, and
    $\overline{\text{lemmas}}_\text{A}$ in D_Q

  return D_S, D_Q
end function

4.3.1 Methods

To test the capacity of cross-transfer to produce an inductive bias for inflectional morphology, the analysis of Chapter 3 is operationalised. The same multi-domain parallel corpus used for estimating the expected risk is used as a training set, with the held-out portion being used as an in domain test set. Furthermore, the same model, a OPUS-MT trained Marian encoder-decoder transformer, is used, validating the use of the estimated mean risk for task sampling. Further evaluation is performed using Meta AI's FLORES corpus as an out-of-domain test set [87]; a set of 2k difficult sentences manually translated to a large number of languages[1].

[87]: Costa-jussà et al. (2022), 'No Language Left Behind: Scaling Human-Centered Machine Translation'

1: At time of use 101, but as of 06-2022, 200

Much like Chapter 3, the morphologically informed loss is computed by feeding in the full source-side text, and preceding target-side context, and requiring the model to predict the target token in its entirety. The full sentence is not used to avoid destroying the learned data distribution[2]. Symbolically, for a single sample with target token y_t, this gives,

2: Particular morphological features in the gathered dataset likely have similar contexts, not all of which is relevant. The model should, obviously, not learn to correlate these features.

$$\mathcal{L}^{(\text{Morph.})}\left(f_t\left(x, y_{<t}; \theta\right), y_t\right) = -\sum_{\tau=1}^{T} \mathbb{1}\left(y_t\right) \log\left(f_t\left(x, y; \theta\right)_\tau\right). \tag{4.4}$$

In practise, a left truncated $y_{\max(1,t-\text{max\_tokens}):<t}$ is fed to the decoder, whereas x is right truncated.

Early experimentation revealed the necessity of including an NMT loss to ensure the NMT system retains its capacity to translate full length texts. The full loss thus becomes,

$$\mathcal{L} = \eta \cdot \mathcal{L}^{(\text{NMT})}\left(f_\theta\left(x, y\right), y\right) + (1-\eta) \cdot \mathcal{L}^{(\text{Morph.})}\left(f_\theta\left(x, y_{<t}\right), y_t\right), \tag{4.5}$$

where $\eta \in [0, 1]$ is a parameter governing the trade-off between the conditional masked language modelling loss and the morphologically informed loss. When applied to meta learning, two distinct formulation of $\mathcal{L}^{(\text{NMT})}$ are considered:

1. **NMT as auxiliary task**: loss is computed as normal, without adaptation. Essentially this yields a multitask learning paradigm, where meta learning with a morphologically informed loss is just another task, and η controls the importance of the two tasks relative to on another.
2. **NMT as meta-regularizer**: loss is computed after adaptation to randomly sampled sentences from the corpus. The purpose of including NMT loss is now specifically to regularize the adaptation step. The morphologically informed and NMT losses are now identical, save for an informed masking in the former (loss and adaptation only occurs for token $y_{<t}$)

During meta-training, due to the memory footprint incurred by the adaptation step, η controls the probability that an episode takes one of the above losses, rather than combining them in parallel. The meta-batch update should, in expectation, approximate Eq. 4.5.

With regards to meta-learning, ANIL [83] was used. The final language-modelling head, projecting the decoder's hidden state output to the target-side vocabulary, was assumed to be the equivalent to the classifier head used for their few-shot experiments. Given the model was already pre-trained, and the proximity of the tasks to the overall dataset, only a single adaptation step was allowed, without first-order approximation. The inner loop learning rate was left as a learnable parameter, specified per step and per layer as suggested by Antoniou, Edwards, and Storkey [88], although in this case that amounts to a single parameter. The outer loop learning was set to $5e-6$, and the initial inner loop learning was set to $1e-3$. To stabilise gradients, a meta-batch size of 8 episodes was chosen, significantly reducing the gradient norm. For cross-transfer episodes, a maximum of 4 lemmas were samples, each with at most 2 samples.

[88]: Antoniou et al. (2018), 'How to train your MAML'

To enable fair comparison, two baseline methods were trained. The first consists of a simple fine-tuning phase, without label smoothing and a linearly decaying learning rate. The second incorporates an explicit morphological signal, namely via training a secondary classifier that predicts the annotated morphological features sets, in the style of the taggers trained in Chapter 2. Layer attention, like presented in `UDIFY`, is applied, allowing the morphological classifier to leverage representations throughout the model. The learning rate was set to $5e-6$, and early stopping was performed after failing to improve the NMT loss on the validation set (sampled prior to training from the training set) in three consecutive checks at 10,000 steps.

In all instances, the models are allowed to train for 1 epoch. In the case of meta-learning, fewer gradient updates are allowed (but with larger batch sizes), although the amount of data seen corresponds to that of 1 epoch. No additional validation set was sampled for meta-learning. Instead, validation was performed by sampling cross-transfer episodes from the training set, and tracking the loss on the relevant and irrelevant (context) tokens pre- and post-adaptation.

Table 4.1: Models evaluated and compared to baselines using popular NMT metrics. The arrows indicate whether larger or smaller values are desired, and ◇ indicates a value of 0 is ideal. Bold values provide the best performing system for the test-set considered, underlined the second-best. Metrics should be compared *within* the test set.

Test Set	Method		η	BLEU↑	BLEU Lemma↑	ΔBLEU◇	ChrF++↑	COMET↑	COMET MQM↑
Ataman Multidomain	**Pretrained**		1.00	29.59	32.80	**3.21**	48.39	0.4471	0.0389
	1 step	Finetuned	1.00	31.27	34.68	3.41	50.29	0.4765	0.0392
		Multitask	0.75	31.33	34.77	3.44	50.29	0.4759	0.0392
		ANIL w/ Cross-Transfer, & NMT Aux.	0.25	30.10	33.52	3.42	49.04	0.4503	0.0386
			0.50	30.34	33.77	3.34	49.45	0.4591	0.0389
			0.75	30.82	34.25	3.43	49.81	0.4668	0.0390
	2 step	Finetuned	1.00	**31.37**	**34.80**	3.43	**50.38**	**0.4810**	**0.0393**
		ANIL w/ Cross-Transfer, & NMT Reg.	0.25	30.90	34.27	3.37	49.96	0.4681	0.0391
			0.50	31.04	34.41	3.37	50.05	0.4732	0.0391
			0.75	31.18	34.57	3.39	50.19	0.4719	0.0391
FLORES	**Pretrained**		1.00	28.82	34.68	5.86	53.90	0.7181	0.0437
	1 step	Finetuned	1.00	29.45	34.69	5.24	54.08	0.7385	0.0447
		Multitask	0.75	**29.53**	34.83	5.30	54.14	0.7331	0.0446
		ANIL w/ Cross-Transfer, & NMT Aux.	0.25	29.24	34.93	5.69	53.92	0.7341	0.0446
			0.50	29.46	35.03	5.57	**54.20**	0.7430	0.0448
			0.75	29.50	34.92	5.42	54.18	**0.7511**	**0.0449**
	2 step	Finetuned	1.00	29.52	34.72	5.20	54.17	0.7431	0.0448
		ANIL w/ Cross-Transfer, & NMT Reg.	0.25	28.37	33.49	**5.12**	53.26	0.7218	0.0444
			0.50	28.78	33.95	5.17	53.56	0.7320	0.0446
			0.75	29.41	**35.27**	5.86	54.00	0.7322	0.0446

The fine-tuning proved to be more successful than anticipated, considering both the in- and out-of-domain test sets, indicating the model was underfit to the training data. Adaptation with cross-transfer was conducted again, but instead of adapting from the pre-trained model, adaptation starts from the fine-tuned model, again for 1 epoch's worth of data. As a baseline, a fine-tuned model is allowed to train for 2 epochs[3]. Meta-learning with NMT as an auxiliary task proved significantly better when meta-learning from the pre-trained baseline, but significantly worse when adapting from the fine-tuned baseline.

3: Similarly to meta-learning, a multi-task learning from the fine-tuned model was attempted. Regardless of learning rate, this consistently led to divergence in the NMT loss.

Much like the experiments presented in Chapter 2, all hyper-parameter tracking was conducted using Weights & Biases [36]. These are publicly available here for the baselines[4] and here for the meta-learning adapted systems[5]. All used code and datasets will be open-sourced [6], and should allow for easy replication.

4: https://wandb.ai/verhivo/nmt_-adapt_baselines

5: https://wandb.ai/verhivo/nmt_-adapt_test

6: https://github.com/IvoOVerhoeven/msc_-thesis

4.3.2 Results

NMT Metrics

Table 4.1 summarises the performance of the systems post-adaptation as a generic NMT system. Baselines are included for comparison. The choice for and interpretation of metrics is provided in Section C.2.

With regards to meta-training with morphological cross-transfer, regardless of the test-set, lower values of η (i.e. lower priority to regular NMT training) result in lower scores. This indicates that the NMT and morphologically informed objectives do not necessarily align, with a significant proportion of regular NMT training required to retain a system capable of functioning as a translator. Generally, performance lags relative to fine-tuning or multi-task training, and when adapting from a fine-tuned model, adaptation has a slightly detrimental effect.

This lag is especially prevalent for the in-domain test set. For the out-of-domain test set, the NMT metrics capable of utilizing sub-word information (`ChrF++`, `COMET` & `COMET - MQM`) 1 stage meta-training with cross-transfer outperforms slightly. The conclusion that models trained in this regime pick up more general word-formation processes is hasty, however, with relatively low `BLEU` score, yet relatively high `BLEU - Lemma` scores as well. The indicate the models perform lexical choice better, but not morphological processing (it chooses the right lemma, but not the right word-form). In turn, the 2 stage meta-trained models with lower values of η, exhibiting low Δ`BLEU` scores, suffer on all other metrics.

Multi-task training achieves higher `BLEU` scores than simply fine-tuning, but the difference largely dissipates when considering the other metrics, or 2 epoch fine-tuning. One potential reason, aside from actually learning representations conductive to improved morphological understanding, is that the combined loss provides a limited level of noise, reducing the speed at which the NMT loss decreases. With early stopping implemented, this leads to slightly longer training times than simply fine-tuning.

All in all, when it comes to general translation, unsurprisingly, fine-tuning proves a difficult baseline to overcome. Turning off label smoothing proved crucial, in both adaptation to the new multi-domain training set, and the harder out-of-domain test set, indicating the model does learn generalizable language concepts.

Morphological Metrics

Whether the adapted systems become more aware of inflectional morphology cannot, generally speaking, be answered by using NMT metrics. Instead, the analysis method introduced in Section 3.2 is repeated. Improvement on the training set is assumed, with test sets instead matching those above. The expected risk per morphological tag set is computed for each model individually, with differences in the expected risk indicating improvement (or not).

Figure 4.8 provides such expected risk differences between systems visually. The text on top provides the mean difference in expected risks in three ways: i) considering all tasks, ii) weighting tasks by the inverse measurement variance, and iii) weighting tasks by the task distribution's probability. The top row of plots provide for all tasks the performance using the baseline and adapted system, with the dashed line indicating no change. Between the training and tests, many morphological tag sets are not present in both. The lighter the colour and the larger the radius of each circle, the higher the probability of sampling that task was in the task distribution. Red dots indicate presence in the test set, but not in the task distribution (i.e. not present in the training set such that it lends itself

to morphological cross-transfer). The lower rows provide histograms of the differences, per class of tasks. The first, red, gives the distribution of tasks not included in the task distribution. The next four segregate seen tasks by the probability of being sampled, chunked into quantiles. Again, brighter colours indicate higher likelihood of being sampled. The final set of histograms instead provide the differences segregated into quantiles of baseline performance (the horizontal axis of the top scatter plot). The lighter the colour, the higher initial scores already were. Figure 4.8 only shows the performance of two meta-trained systems relative to 1 stage fine-tuning, and only in terms of morph tag set IoU. Section C.3 provides additional system comparisons, including fine-tuning to pre-trained and multi-task to fine-tuning, these provide a baseline comparison for expected morphological competence improvement. Also provided are similar figures but considering character driven metrics (e.g. Levenshtein distance).

Despite showing diminished NMT metric values, two stage meta-training ($\eta = 0.50$) does show definite improvement in the expected morph tag set IoUs, comparable in size to fine tuning from the pre-trained baseline system (see Figure C.4). On tasks included in the adaptation phase, the model shows a 0.021 & 0.05 increase on the ID and OOD test sets, respectively. The effect of the task distribution is also apparent, with tasks in the higher likelihood quantiles scoring higher than those in the lower ones. Unscheduled tasks, instead, seem to be distributed about the null point. The noted effect is especially prevalent when comparing to Figure C.4; here there is a small but steady increase, there all distributions and medians/means align almost perfectly across quantiles. Unfortunately, likely due to the task distribution being defined in terms of high rates of confusion, improvement occurs mostly in tasks already achieving high scores. The 25th percentile tasks are distributed about the null point much like the unscheduled tasks, and for the OOD test set, this also extends to 50th percentile. Thus, tasks where the model already performed well saw the brunt of improvement rather than those where it performed poorly. The degree to which these changes are caused by the meta-training can be seen when comparing to lower and higher values of η, see Figure C.6. When $\eta = 0.25$, the effect is more noticeable, even for the OOD test set, while $\eta = 0.75$ leads to distributions similar to those without morphological cross-transfer training. Thus, η effectively trades-off general NMT capacity for inflectional capacity.

The same cannot be said for 1 stage meta-learning. Despite competitive sub-word level NMT scores on the OOD test set, and a slight lag on the in-domain set, with respect to expected morphological risk, the models lag considerably. Perhaps more worryingly, the task distribution seems to have no tangible effect, with improvement or loss matching the ignored tasks in distribution. It seems, NMT training as an auxiliary task dominates the morphologically informed training, booking improvement on the former while showing a detrimental effect in the latter.

It is important to note that the presented effect sizes are small, and while visible, suffer from high variance. Rigorous claims as to improvement are, unfortunately, difficult to make.

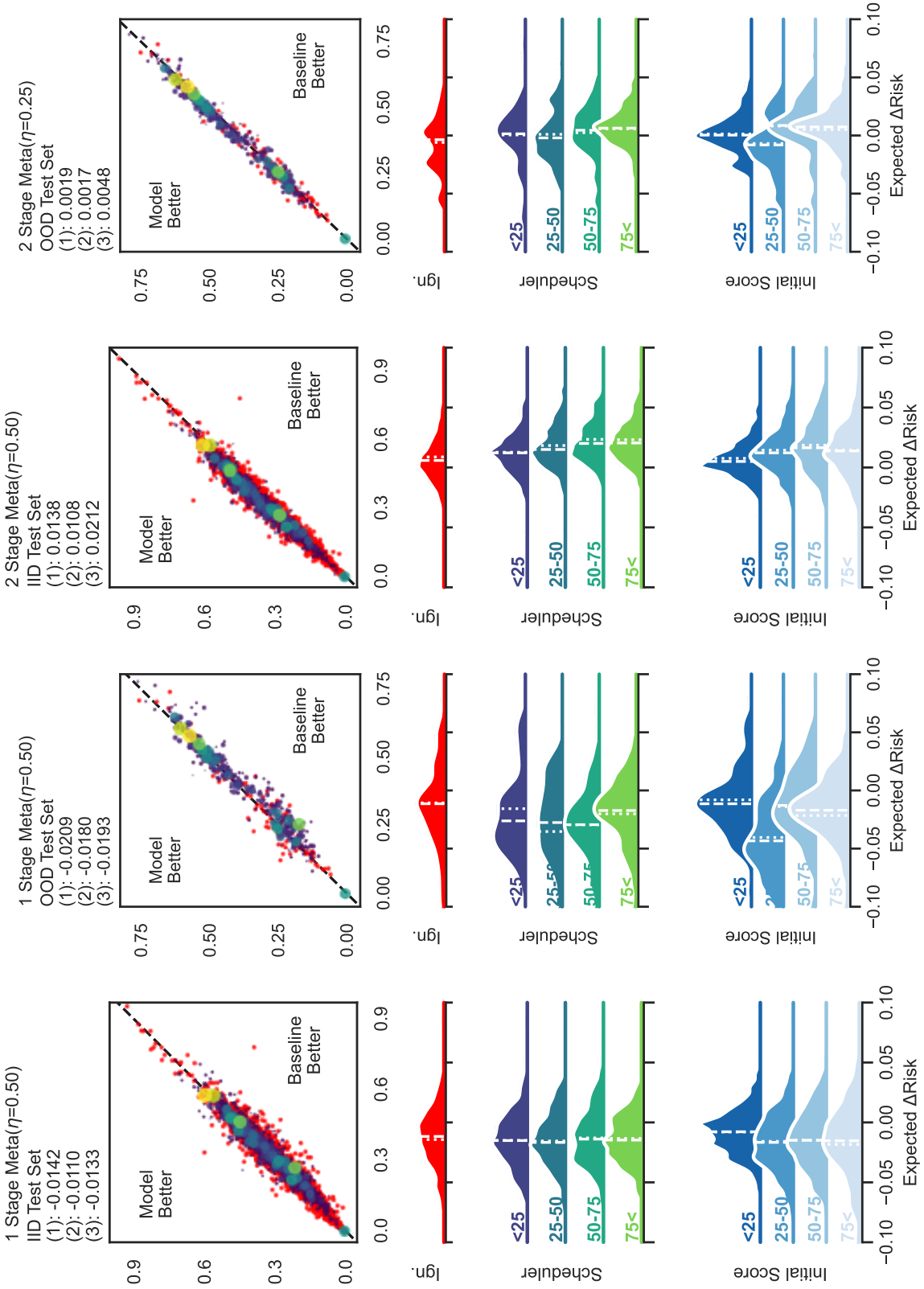


Figure 4.8: Figures testing the morphological competence of adapted systems relative to the 1 stage fine-tuned baseline.

4.4 Discussion

The initially defined task of inducing morphological awareness in pre-trained NMT systems has finally been addressed. Where existing methods do so by either altering the data or how the model sees the data, the methodology presented here manages to avoid all of these, retaining an NMT system capable of translation. By utilising the risk estimation technique discussed in Chapter 3, a slight improvement in morphological competence can be seen, with the improvement increasing as more emphasis is placed on morphological cross-transfer. Unfortunately, both general NMT quality and morphological competence do not occur together. Instead, an increase in the latter comes at a decrease in the former, whereas an increase in the former sees a global increase in the latter.

Thus, morphological cross-transfer shows an increase in morphological competence metrics, both on tasks included in the scheduler and to a limited extent also on tasks not included. The reduction in NMT quality evaluation likely means the increased awareness of context yielding certain morphological features results in neglecting other contexts. One potential cause is a misspecified task sampling distribution. Given morphological cross-transfer requires word-form switching of lemmas in the support and query sets, using confusion as the metric guiding the task sampling seemed natural. Using the risk estimates instead, might lead to more direct increases of the morphological competence at lower risk of reducing translation quality. Otherwise, it might be necessary to relax the requirement of constant lemmas in both support and query sets, instead only holding the morphological feature set constant. This is more akin to standard GBML, treating a morphological feature set as an individual task. This no longer teaches 'copy-and-inflect', however, instead just an association between morphological features and likely word-forms. This has the added benefit of providing far more data, allowing lemmas included that inflect in totally different ways (for example, irregular forms) although how useful this remains to be seen.

Conclusion 5

Presented in this thesis is a new technique for teaching an inductive bias for morphological inflection, and an extensive analysis of the necessary ingredients for it. Throughout, each method was placed in the context of existing literature, and novel contributions were made clear. Potential issues with the presented methodologies were highlighted, and future research directions briefly touched upon. While successful to some degree, there clearly remains much to do before true morphological competence is achieved.

To call neural language modelling a rapidly developing field is an egregious understatement. State-of-the-art techniques are published often, opening up new tasks and new benchmarks, before these too are quickly overcome. Within NMT, but also generally, scaling existing techniques seems to be a popular and successful method for improving translation quality for many languages. A prime example of this is MetaAI's 'No Language Left Behind' project [89], recently publishing results of a model capable of scaling to 200 languages. While showing some architectural innovations, most effort seems to have been spent on gathering more and better quality data, larger parameter sets and better evaluation. Despite unseen scores on many languages, a substantial number of which clearly fall in the low-resource category, there remains a gap between the morphologically poor and morphologically rich languages. The presented model counts 12 billion parameters, roughly 100 times more than the models used in Chapters 3 and C. Relative to other state-of-the-art language models, this is still considered small. For research into the morphological awareness of neural language models, thus, designing viable post-hoc techniques thus become more and more important. Designing specialized architectures for this task, or focusing on language-specific efforts, will only serve to increase the rift between academia and industry.

The topics covered in this thesis primarily built on ideas and methods implemented with smaller-scale (recurrent) architectures. While undeniably important, prior work could afford pre-training with custom built architectures, or increasing the computational complexity of the training phase for their task. This work, instead, can be seen to make an attempt at bringing research closer to the current state of affairs in NMT research. Building model- and language-agnostic adaptation techniques will likely become increasingly vital for capturing the wide variety of linguistic phenomena present in human language. With regards to morphological awareness, better automated taggers and lemmatizers, and a better evaluation method will be necessary components for future methods. Morphological cross-transfer, is only one such method, and has shown some indication of achieving what it sets out. Integrating it with standard NMT training will prove paramount for future success.

Appendix

Morphological Tagging and Lemmatization in Context

A

Table A.1: Language merged UD treebanks, filtered by having at least 1 start of quality. Gives the number of constituent treebanks, the number of sentences (thousands), the number of tokens (thousand), the length of the set of genres present in the treebanks, and the average quality in stars.

Language	Family	Treebanks	Sentences (k)	Tokens (k)	Genres	Stars
Afrikaans	IE, Germanic	1	2	49	2	3.5
Arabic	Afro-Asiatic, Semitic	1	8	242	1	3.0
Armenian	IE, Armenian	1	3	52	6	4.0
Belarusian	IE, Slavic	1	25	305	7	4.5
Bulgarian	IE, Slavic	1	11	156	3	4.0
Catalan	IE, Romance	1	17	537	1	4.0
Croatian	IE, Slavic	1	9	199	3	4.0
Czech	IE, Slavic	4	127	2204	5	4.0
Dutch	IE, Germanic	2	21	307	1	2.5
English	IE, Germanic	6	38	608	2	3.0
Estonian	Uralic, Finnic	2	37	511	4	4.0
Finnish	Uralic, Finnic	1	15	202	6	3.5
French	IE, Romance	5	25	559	4	3.5
Galician	IE, Romance	1	1	23	1	3.5
German	IE, Germanic	2	206	3687	3	4.0
Greek	IE, Greek	1	3	62	3	3.5
Icelandic	IE, Germanic	2	51	1142	4	3.0
Indonesian	Austronesian, Malayo-Sumbawan	2	7	148	2	3.5
Irish	IE, Celtic	1	5	116	5	2.0
Italian	IE, Romance	5	34	737	3	3.5
Japanese	Japanese	2	16	344	2	2.0
Latin	IE, Latin	1	9	242	2	4.0
Latvian	IE, Baltic	1	16	266	5	3.5
Lithuanian	IE, Baltic	2	4	75	4	2.5
Norwegian	IE, Germanic	2	38	612	3	4.0
Polish	IE, Slavic	2	39	478	5	4.0
Portuguese	IE, Romance	1	9	211	1	4.0
Romanian	IE, Romance	3	40	937	2	4.0
Russian	IE, Slavic	3	110	1813	1	4.0
Serbian	IE, Slavic	1	4	98	1	4.0
Slovak	IE, Slavic	1	11	106	3	3.5
Slovenian	IE, Slavic	2	11	170	3	3.5
Spanish	IE, Romance	1	18	555	1	4.0
Swedish	IE, Germanic	1	5	91	3	3.5
Tamil	Dravidian, Southern	1	1	9	1	2.5
Telugu	Dravidian, South Central	1	1	6	1	1.0
Turkish	Turkic, Southwestern	6	73	640	2	3.5
Welsh	IE, Celtic	1	2	41	5	2.5

Table A.2: Multilingual pre-training with monolingual fine-tuning.

Model	Language	Lemma Acc.	Lev. Dist.	Morph. Set Acc.	Morph. Tag F1	Throughput
UDIFY Multi+Mono	Arabic	0.94	0.17	0.93	0.97/0.88	2313
	Dutch	0.96	0.08	0.96	0.97/0.97	2507
	English	0.97	0.05	0.93	0.96/0.90	2445
	Finnish	0.91	0.19	0.93	0.97/0.89	1915
	Turkish	0.94	0.13	0.83	0.92/0.73	1407
	Mono. Mean	0.93	0.16	0.89	0.94/0.85	2113
	Mean	0.94	0.12	0.92	0.96/0.87	2117
DogTag Multi+Mono	Arabic	0.93	0.20	0.88	0.94/0.84	1851
	Finnish	0.87	0.29	0.86	0.94/0.83	2106
	Turkish	0.91	0.19	0.78	0.90/0.72	1714
	Mono. Mean	0.90	0.23	0.83	0.93/0.79	1827
	Mean	0.90	0.23	0.84	0.93/0.80	1890

Evaluating the Morphological Awareness of NMT Systems

B

B.1 Identifying Problematic Morphological Features, Cont.

The very large Table B.1 provides the model averaged posterior coefficients of the extended model. The category and subcategory (the morphological dimension and tag) are provided in the left two columns, with estimates being divided into relevant parts-of-speech. Hidden are estimates for the category being omitted entirely, which (given the possible category-PoS combinations are known a priori) correspond to malformed or words not recognized by the tagger. Already, far more information is provided, evidenced by a substantial gain in explained variance ($\Delta R^2 = 0.053$), and the best model being $4.90e + 37$ times more likely than only considering parts-of-speech.

The interpretation of the global part-of-speech effects are slightly altered now. These are partial effects, with much of the explanatory power being shifted to the feature effects. For example, when considering nouns, β_{noun} averages out to 0, implying that nouns cannot be differentiated in the data from any other part-of-speech, without considering the morphological features present. To get the expected risk for a feminine pronoun, as another example, regardless of other morphological features, the effects of the intercept, the part-of-speech and tag effect must all be summed: in this case, $0.268 - 0.15 + 0.053 = 0.171$.

For some morphological categories, a discernible difference exists across tags. For most forms of casing, the effect is detrimental, with the magnitude of the effect being relatively stable across parts-of-speech. An exemption to this is the 'Nominative' case[1], with all estimates being the highest for that part-of-speech in the casing category (save the adjectives). In turn, the 'Vocative' case[2], which is relatively rare and only present for nouns, sees the lowest weight, with a dismal expected IoU of 0.03. Second lowest for the nouns, and typically lowest for all other parts-of-speech also, is the 'Instrumental' case[3].

Another category with clear differences is that of 'Gender & Animacy'. The easiest gender to inflect correctly appears to be either 'Feminine' or 'Neuter', especially for proper nouns and pronouns, respectively. Determiners marked for gender tend to perform poorly regardless, indicating that matching with the noun it modifies is difficult. Finally, a clear ordering exists for the 'Person' category, with capacity to inflect decreasing steadily going from 1st to 3rd person.

1: Identifying the word as the subject of a verb.

2: Identifying the word, typically a person, as being addressed

3: Identifying the word as the instrument by which an action is taken

Table B.1: Posterior of the dependent variables weights, resulting from a Bayesian linear regression for the IoU of the predicted and ground-truth morphological tag sets. Uses same methodology as Table 3.1, which is also the null model.

Category	Subcategory	Part-of-Speech	p(incl | data)	Mean	SD	95 CI LB	95 CI UB
Intercept			1.00	0.268	0.000	0.267	0.268
Parts-of-Speech		Adjective	0.00	0.000	0.000	0.000	0.000
		Participle (Adj)	1.00	−0.080	0.005	−0.091	−0.070
		Adposition	1.00	0.043	0.003	0.037	0.049
		Adverb	0.00	0.000	0.000	0.000	0.000
		Determiner	1.00	−0.123	0.004	−0.131	−0.115
		Noun	0.00	0.000	0.000	0.000	0.000
		Numeral	1.00	0.015	0.005	0.005	0.025
		Pronoun	1.00	−0.150	0.005	−0.160	−0.142
		Proper Noun	0.00	0.000	0.000	0.000	0.000
		Verb	1.00	−0.076	0.005	−0.085	−0.067
		Participle (Verb)	1.00	−0.064	0.015	−0.096	−0.035
Casing	**Dative**	Adjective	1.00	−0.021	0.003	−0.027	−0.016
		Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Determiner	0.00	0.000	0.000	0.000	0.000
		Noun	1.00	−0.012	0.003	−0.019	−0.006
		Numeral	1.00	−0.073	0.008	−0.089	−0.057
		Pronoun	1.00	−0.068	0.003	−0.075	−0.062
		Proper Noun	1.00	−0.018	0.007	−0.031	−0.006
	Essive	Adjective	1.00	0.001	0.003	−0.004	0.006
		Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Determiner	1.00	0.019	0.003	0.014	0.024
		Noun	1.00	−0.021	0.003	−0.027	−0.015
		Numeral	1.00	−0.049	0.006	−0.061	−0.037
		Pronoun	1.00	−0.097	0.004	−0.104	−0.090
		Proper Noun	1.00	−0.046	0.006	−0.057	−0.035
	Genitive	Adjective	1.00	−0.063	0.002	−0.067	−0.059
		Participle (Adj)	1.00	−0.048	0.004	−0.055	−0.041
		Determiner	1.00	−0.015	0.002	−0.020	−0.010
		Noun	1.00	−0.027	0.003	−0.032	−0.022
		Numeral	1.00	−0.070	0.005	−0.080	−0.061
		Pronoun	1.00	−0.079	0.003	−0.086	−0.073
		Proper Noun	1.00	−0.038	0.005	−0.049	−0.028
	Instrumental	Adjective	1.00	−0.022	0.002	−0.027	−0.018
		Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Determiner	0.00	0.000	0.000	0.000	0.000
		Noun	1.00	−0.049	0.003	−0.055	−0.043
		Numeral	0.00	0.000	0.000	0.000	0.000
		Pronoun	1.00	−0.114	0.003	−0.120	−0.108
		Proper Noun	1.00	−0.048	0.006	−0.060	−0.036
	Nominative	Adjective	1.00	−0.024	0.002	−0.028	−0.020
		Participle (Adj)	1.00	−0.020	0.004	−0.028	−0.013
		Determiner	1.00	0.059	0.002	0.055	0.062
		Noun	1.00	0.032	0.003	0.027	0.037
		Numeral	1.00	0.016	0.004	0.007	0.024

continues on the next page

Category	Subcategory	Part-of-Speech	p(incl | data)	Mean	SD	95 CI LB	95 CI UB
Casing	**Nominative**	Pronoun	1.00	−0.057	0.003	−0.062	−0.051
		Proper Noun	1.00	0.019	0.005	0.009	0.028
	Vocative	Noun	1.00	−0.237	0.012	−0.260	−0.214
Gender & Animacy	**Feminine**	Determiner	1.00	−0.112	0.006	−0.125	−0.100
		Noun	0.00	0.000	0.000	0.000	0.000
		Pronoun	0.00	−0.000	0.000	0.000	0.000
		Proper Noun	1.00	0.053	0.004	0.044	0.062
	Masculine	Determiner	1.00	−0.139	0.007	−0.153	−0.127
		Noun	0.00	−0.000	0.000	0.000	0.000
		Pronoun	0.00	0.000	0.000	0.000	0.000
		Proper Noun	1.00	0.015	0.005	0.005	0.024
	Neuter	Determiner	1.00	−0.135	0.007	−0.149	−0.122
		Noun	1.00	0.019	0.002	0.014	0.023
		Pronoun	1.00	0.049	0.005	0.038	0.059
		Proper Noun	0.00	0.000	0.000	0.000	0.000
	Animate	Determiner	0.00	0.000	0.000	0.000	0.000
		Noun	0.00	0.000	0.000	0.000	0.000
		Pronoun	1.00	−0.050	0.005	−0.059	−0.040
		Proper Noun	1.00	0.040	0.004	0.032	0.047
	Inanimate	Determiner	1.00	0.006	0.004	−0.002	0.013
		Noun	1.00	0.024	0.003	0.018	0.030
		Pronoun	0.00	0.000	0.000	0.000	0.000
		Proper Noun	0.00	−0.000	0.000	0.000	0.000
Number	**Singular**	Adjective	1.00	0.022	0.003	0.017	0.027
		Participle (Adj)	1.00	0.016	0.003	0.010	0.020
		Determiner	1.00	0.010	0.002	0.006	0.013
		Noun	0.00	−0.000	0.000	0.000	0.000
		Pronoun	1.00	−0.049	0.012	−0.071	−0.027
		Proper Noun	0.00	0.000	0.000	0.000	0.000
		Verb	0.00	0.000	0.000	0.000	0.000
		Participle (Verb)	1.00	0.052	0.003	0.045	0.058
	Plural	Adjective	1.00	0.019	0.003	0.014	0.024
		Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Determiner	0.00	−0.000	0.000	0.000	0.000
		Noun	1.00	0.021	0.002	0.018	0.025
		Pronoun	1.00	−0.014	0.012	−0.037	0.008
		Proper Noun	1.00	−0.027	0.004	−0.035	−0.019
		Verb	1.00	−0.018	0.002	−0.022	−0.014
		Participle (Verb)	0.00	0.000	0.000	0.000	0.000
Verbal	**Perfective**	Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Verb	1.00	−0.009	0.007	−0.024	0.004
		Participle (Verb)	1.00	−0.046	0.003	−0.053	−0.040
	Imperfective	Participle (Adj)	1.00	−0.018	0.005	−0.027	−0.009
		Verb	1.00	−0.006	0.007	−0.021	0.007
		Participle (Verb)	1.00	−0.052	0.003	−0.058	−0.045
	Finite	Verb	1.00	−0.067	0.006	−0.078	−0.056

continues on the next page

Category	Subcategory	Part-of-Speech	p(incl | data)	Mean	SD	95 CI LB	95 CI UB
Verbal	Nonfinite	Verb	0.00	0.000	0.000	0.000	0.000
	Conditional	Verb	1.00	−0.109	0.007	−0.123	−0.096
	Imp.-Jussive	Verb	1.00	−0.029	0.006	−0.040	−0.019
	Indicative	Verb	1.00	−0.052	0.008	−0.069	−0.036
	Present Tense	Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Verb	1.00	−0.023	0.006	−0.036	−0.011
		Participle (Verb)	0.00	0.000	0.000	0.000	0.000
	Past Tense	Participle (Verb)	1.00	0.029	0.015	−0.001	0.057
	Fut. Tense	Verb	1.00	−0.052	0.007	−0.066	−0.039
	Active	Participle (Adj)	0.00	0.000	0.000	0.000	0.000
		Verb	1.00	0.030	0.005	0.018	0.035
		Participle (Verb)	0.00	0.000	0.000	0.000	0.000
	Passive	Participle (Adj)	0.00	0.000	0.000	0.000	0.000
Comparison	Comp.	Adjective	1.00	0.020	0.002	0.016	0.023
		Adverb	1.00	−0.053	0.007	−0.067	−0.039
	Rel.	Adjective	1.00	0.113	0.002	0.108	0.117
		Adverb	0.00	0.000	0.000	0.000	0.000
Person	1st	Determiner	0.00	0.000	0.000	0.000	0.000
		Pronoun	1.00	0.076	0.004	0.068	0.082
		Verb	1.00	−0.011	0.004	−0.018	−0.004
	2nd	Determiner	1.00	−0.027	0.003	−0.033	−0.021
		Pronoun	1.00	0.034	0.004	0.026	0.041
		Verb	1.00	−0.041	0.004	−0.049	−0.034
	3rd	Determiner	1.00	−0.066	0.003	−0.073	−0.061
		Pronoun	0.00	0.000	0.000	0.000	0.000
		Verb	1.00	−0.102	0.005	−0.111	−0.093
Observations		510,778					
R^2		0.142					
$p(M^{(\text{null})}\|\text{Data})$		0.000					
$p(M^{(\text{best})}\|\text{Data})$		0.987					

The NMT system seems to prefer superlatives over their comparative counterparts (e.g. greatest over greater), here represented by adjectives and adverbs marked as 'Rel.' 'Comp.' in the 'Comparison' dimension. In fact, superlative adjectives attract the largest estimated coefficient, with the reduction caused by moving to comparatives (0.093) being larger than most other coefficients.

Taking all information into account, the most problematic parts-of-speech for this NMT system remain the pronouns and determiners. Again, much like the PoS only model, nouns and adjectives perform best of all, especially with certain features being present.

While this analysis allows for identifying troublesome morphological features for conditional generation, despite high variance, it conflates

mistranslations or malformed generations with the capacity to morphologically inflect. For example, a different interpretation in word order between the NMT system and the ground truth translation yields high risk, despite a properly inflected word possibly being generated later in the sentence. While one could limit the estimates to instances where the model was capable of inferring the correct lemma, ensuring weights indicate only the capacity of the NMT system to inflect said lexical items (occurs in roughly 36% of instances), this does not inform the user as to which morphological markers are erroneously produced instead. Precisely this notion is captured by the confusion matrices presented in Subsection 3.3.2.

B.2 Generating a Task Distribution

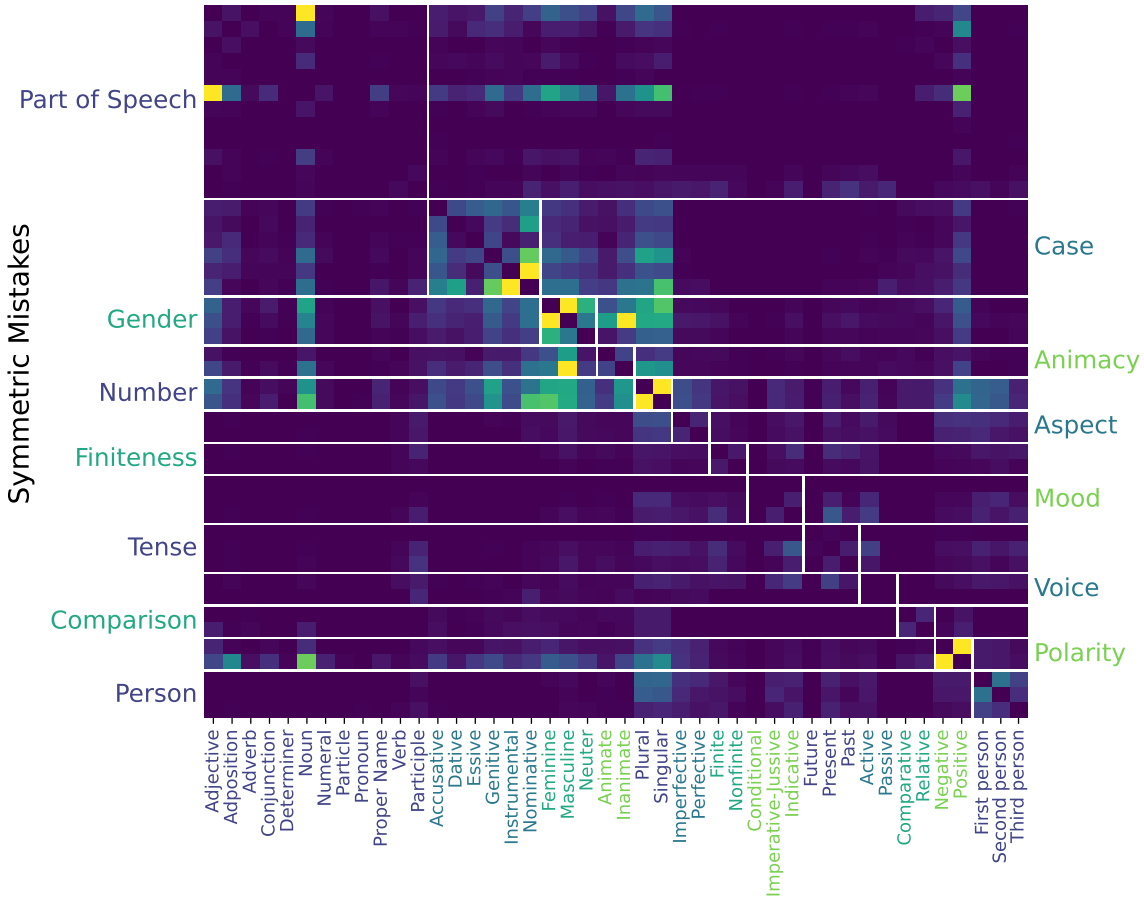


Figure B.1: The mistakes, without considering the difference between ground-truth and predicted, and normalized over all values. In essence, this just provides one with a notion of 'these feature are often confused with each other'. Visually, this is the task distribution as defined in Chapter 4, except marginalised from morphological tag sets to individual tags.

Adapting NMT Systems for Morphological Awareness

C

C.1 Meta-learning & Morphological Cross Transfer

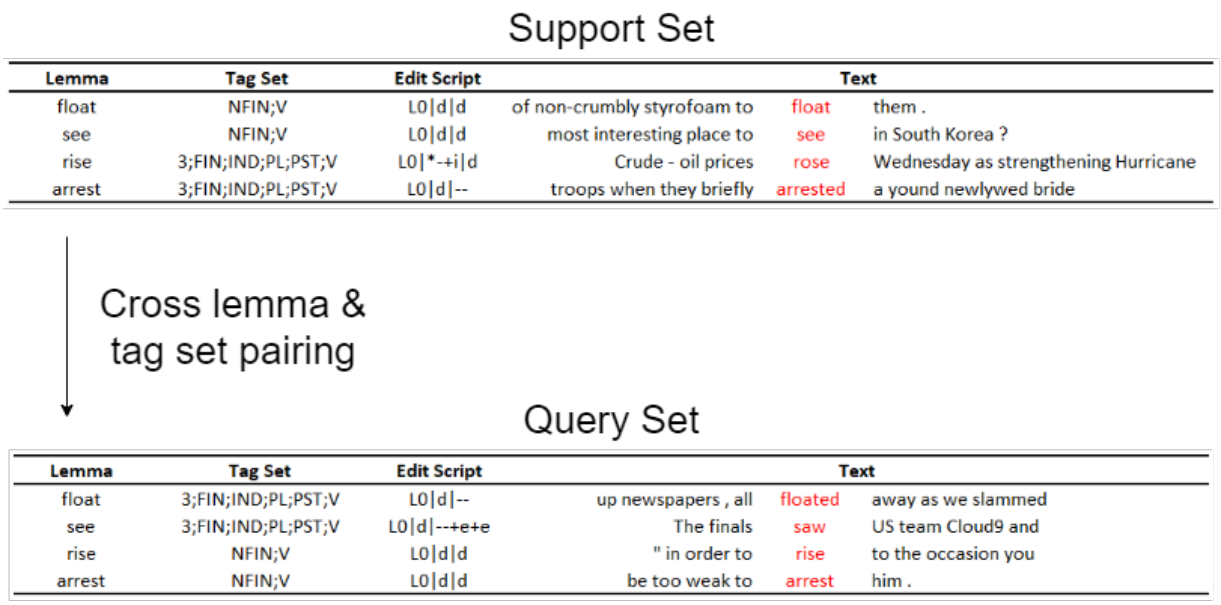


Figure C.1: Verb inflection from finite to plural past tense as sampled using morphological cross-transfer.

Figure C.1 provides, in a nutshell, the core idea of morphological cross-transfer. Two morphological tag sets are sampled as tasks, without holding the lemma edit script constant, and a set of overlapping lemmas are sampled also. Half the lemmas are present in the first task in the support set, whereas the other half of the lemmas are present in the other. In the query set, the roles flip. The red text highlights the word form in a sentence. In practise, the model is only allowed to see preceding context, but also has access to the entire sentence via the encoder. For successful transfer from support set adaptation to query set generalization, the NMT system must recognize from context the appropriate morphological task, and extend the corresponding word-formation processes to lemmas it has already seen, but in new word-forms.

C.1.1 Generalized GBML

Algorithm 2 GBML Iteration with Cross-transfer and Layer-wise Gradient Modulation

for $t = 1, \ldots,$ #episodes per iteration **do**
  Sample $D_{\mathrm{S}}, D_{\mathrm{Q}} \sim \textsc{CrossTransfer}(p\left(\textsc{Task}_i, \textsc{Task}_j\right), \mathcal{D})$
  Generate task model $\theta_0^{(\text{epsiode})} \leftarrow \theta^{(\text{meta})}$
  for $k = 1, \ldots,$ #shots per episode **do**
    Compute $\mathcal{L}_\tau \left(\theta_k^{(\text{epsiode})}, D_{\mathrm{S}}\right)$
    Update $\theta_{k+1}^{(\text{epsiode})} \leftarrow \theta_k^{(\text{epsiode})} - \alpha \nabla_{\theta_k^{(\text{epsiode})}} \mathcal{L}\left(\theta_k^{(\text{epsiode})}, D_{\mathrm{S}}\right)$
  end for
  Compute $\mathcal{L}_\tau \left(\theta_K^{(\text{epsiode})}, D_{\mathrm{Q}}\right)$
end for
Update $\theta \leftarrow \theta + \textsc{Optimizer}\left(\nabla_\theta \sum_\tau \mathcal{L}_\tau \left(\theta_k^{(\text{epsiode})}, D_{\mathrm{Q}}\right), \beta\right)$
Update $\alpha \leftarrow \alpha + \textsc{Optimizer}\left(\nabla_\alpha \sum_\tau \mathcal{L}_\tau \left(\theta_k^{(\text{epsiode})}, D_{\mathrm{Q}}\right), \beta\right)$

C.1.2 MAML, ANIL & BOIL: Feature Reuse or Fast Adaptation

Again, the goal of GBML is to find a set of weights that simultaneously provide strong initialization for many related tasks, but also one that can adapt quickly to any individual task. Which of these properties dominates, or whether both are present, remains open to debate. Of the two, feature reuse is most relevant to this thesis; per definition it yields higher zero-shot generalization.

In their review, Raghu et al. [83, -5em] find that MAML primarily search for features general enough to be reused across tasks. When ablating the task-specific model to adapt only the classification head, performance degradation is negligible. Even when earlier layers are allowed to adapt also, representation difference between the meta- and episode-model dissipates in earlier layers. Otherwise, for MAML trained models, task-agnostic inductive bias is encoded in the earliest layers, with task-specific features existing exclusively in the final layers. While a finding general to deep learning models, it lead Raghu et al. to propose the Almost No Inner Loop (ANIL) variant of MAML. It strictly enforces feature reuse; only the head is allowed to adapt, while the model's 'body' remains task agnostic. A further benefit is the vastly diminished computational cost of the inner loop, with typically only a minor fraction of the model parameters needing to adapt.

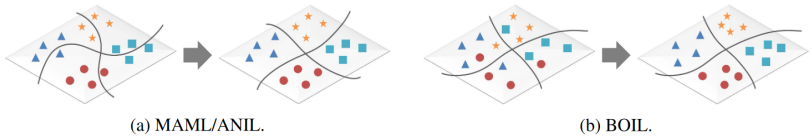
[83]: Raghu et al. (2020), 'Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML'

Arnold, Iqbal, and Sha [90] largely corroborate the finding that depth is important in GBML, with model performance drastically improving with interleaved linear layers. They posit that extending depth allows for earlier layers to generalize, while later layers add specialization. Where their analysis differs, however, is the determination that later layers also aid fast adaptation. The gradients passed to earlier layers are modulated by the values of later layers[1], implying that for successful meta-learning, the primary function of the final layers in the meta-model is to help earlier layers quickly adapt.

[90]: Arnold et al. (2021), 'When maml can adapt fast and how to assist when it cannot'

1: This finding follows simply from the derivative's product rule:

$$\frac{\mathrm{d}}{\mathrm{d}x} f_l(f_{l-1}(x)) = f_l'(f_{l-1}(x)) \cdot f_{l-1}'(x)$$



(a) MAML/ANIL. (b) BOIL.

Figure C.2: Conceptually, the difference between MAML/ANIL (a) based techniques and BOIL (b). Where the decision boundary rapidly shifts in (a), with changes in the features being deferred, (b) shows the exact opposite behaviour. Taken from [91].

Starkly contrasted to ANIL, and taking the findings of Arnold, Iqbal, and Sha to an extreme, Oh et al. [91] propose Body Only Inner Loop (BOIL). Here, the classification head is kept fixed, with only earlier layers being allowed to adapt. Essentially, the final layer's only function is to warp the gradients passed to earlier layers in the model. Thus, using the converse of the argument presented by Raghu et al., BOIL induces fast adaptation of the representations. Their findings show that while earlier layers still exhibit feature-reuse to some extent, only the penultimate layer alters drastically during inner-loop adaptation. In various N-way, K-shot experiments, BOIL proved preferable to MAML/ANIL from 1-shot onwards.

[91]: Oh et al. (2021), '{BOIL}: Towards Representation Change for Few-shot Learning'

In an effort to settle the debate of feature reuse versus adaption, Oswald et al. [92] endow a MAML meta-learner with an updateable parameter-specific binary mask. Rather than a priori freezing a subset of the model's layers, whether a layer is part of the inner loop is simply a parameter that is learned along with the model. In effect, the model self-regulates the sparsity of its updates. Much like previous papers, they find that as learning progresses, earlier layers are adapted less, with only the later layers seeing dense updates. The induced sparsity has the additional benefit of regularization, allowing the underlying model to generalize outside seen domains, while inner loop learning rates and steps can be easily accommodated without worry of overfit. Ultimately, sparse-MAML is capable of outperforming not just MAML variants with adjusted inner-loop regimens, e.g. ANIL or BOIL, but also ones equipped with additional meta-hyper-optimization modules. The latter result is especially surprising, as sparse-MAML is strictly less expressive.

[92]: Oswald et al. (2021), 'Learning where to learn: Gradient sparsity in meta and continual learning'

C.2 NMT Metrics

Section C.2 makes use of a number of NMT metrics, some of which are literature standards, others which are not. This subsection provides some motivation for their inclusion, and some intuition as to their interpretation.

- **BLEU**: since its introduction in 2002, and despite its many flaws, `BLEU` [93] remains ubiquitous. Based on some hypothesis $\tilde{y}$ and the ground-truth reference translation y, it assigns a score between 0 and 1 indicating, roughly, the proportion of n-grams present in y also present in $\tilde{y}$. The metric is computed per sentence, although its essentially meaningless at this level, then aggregated by taking the geometric mean of the sentence values, with short sentences penalized. It is important to note that language needs to be pre-tokenized before scoring, and it operates exclusively at the word-level. To cite Kocmi et al. [94]'s recent review ,

[93]: Papineni et al. (2002), 'Bleu: a method for automatic evaluation of machine translation'

[94]: Kocmi et al. (2021), 'To ship or not to ship: An extensive evaluation of automatic metrics for machine translation.'

> Do not use BLEU, it is inferior to other metrics, and it has been overused.

To that end, inclusion here is only done out of historic consideration, allowing comparison to prior literature (like Ataman, Aziz, and Birch [7]). Post's [95] `SacreBLEU` implementation was used for all BLEU values, a standardized set of practises allowing comparison to literature

[95]: Post (2018), 'A Call for Clarity in Reporting BLEU Scores'

- **BLEU - Lemma**: identical to the above point, but the translations are lemmatized prior to scoring. This removes all morphological features and essentially captures how capable the NMT system is at handling non-morphological language phenomena (e.g. lexical choice, word order, syntax, etc.)
- **ΔBLEU**: the difference of the previous two points. This should answer the converse of the previous point: to which extent do lemma and word-form generation differ. If the value is close to 0, it should indicate little trouble with inflecting the produced lemmas
- **ChrF++**: of all model free NMT metrics, Popović's [96, 97] `ChrF++` is typically tauted as the best. Unlike `BLEU`, sentence scores are computed using both word and character n-grams, and the Fβ score is computed instead, balancing precision and recall ($\beta = 2$ is suggested, implying recall is deemed twice as important as precision). It clearly has access to sub-word information, and increases should indicate systems better capable of handling morphology. Kocmi et al. [94] find it correlates well with human judgement, especially relative to other 'string-based' metrics. Again, `SacreBLEU` is used.

[96]: Popović (2017), 'chrF++: words helping character n-grams'
[97]: Popović (2015), 'chrF: character n-gram F-score for automatic MT evaluation'

- **COMET & COMET MQM**: unlike previous metrics, Rei et al. [98]'s `COMET` computes scores using a pre-trained language model, fine-tuned to predict human judgement scores when fed the source-side and target-side reference translation, along with the candiate translation. Since its introduction, it has been ranked among the top performing systems in both the WMT'20 [99] and WMT'21 [100] shared tasks, and was Kocmi et al.'s best performing metric. When referring to just `COMET`, this is a version built by regressing WMT news task corpora annotated with direct assessments (human judged comparisons of similarity between sentences on an analog scale). `COMET - MQM`, on the other hand, uses data annotated with the MQM grading scheme, which includes several categories of possible errors, of which spelling is one [101]. This allows, to some extent, leveraging sub-word information.
  Both systems do not express absolute scores, but instead unbounded z-values of a model's quality assessment. For `COMET` typical scores fall in the -1 to 1.5 range, while `COMET - MQM` falls within ± 0.4. This is, however, highly dependent on the language pair and the evaluation dataset. Comparison across metrics or across test sets, thus, is not recommended.

[98]: Rei et al. (2020), 'COMET: A Neural Framework for MT Evaluation'

C.3 Additional Experimental Results

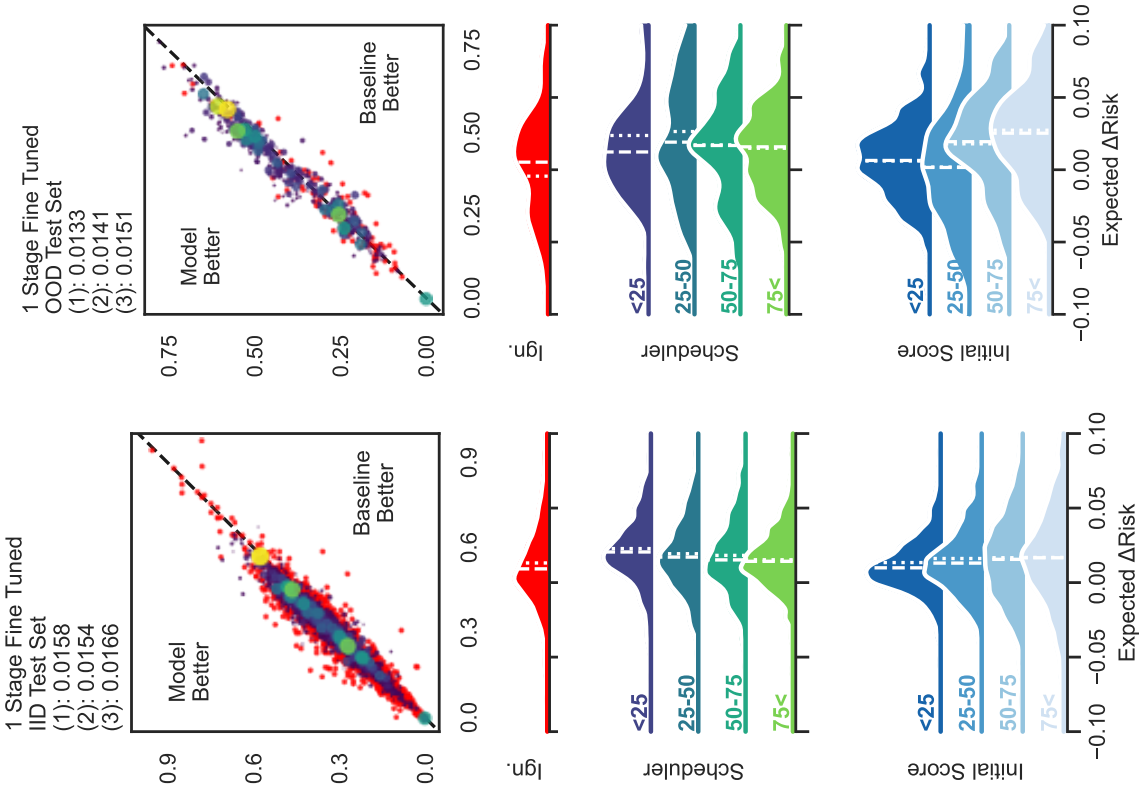


Figure C.3: In the style of Figure 4.8, but now presenting the 1 stage fine-tuned model versus the pre-trained model as baseline.

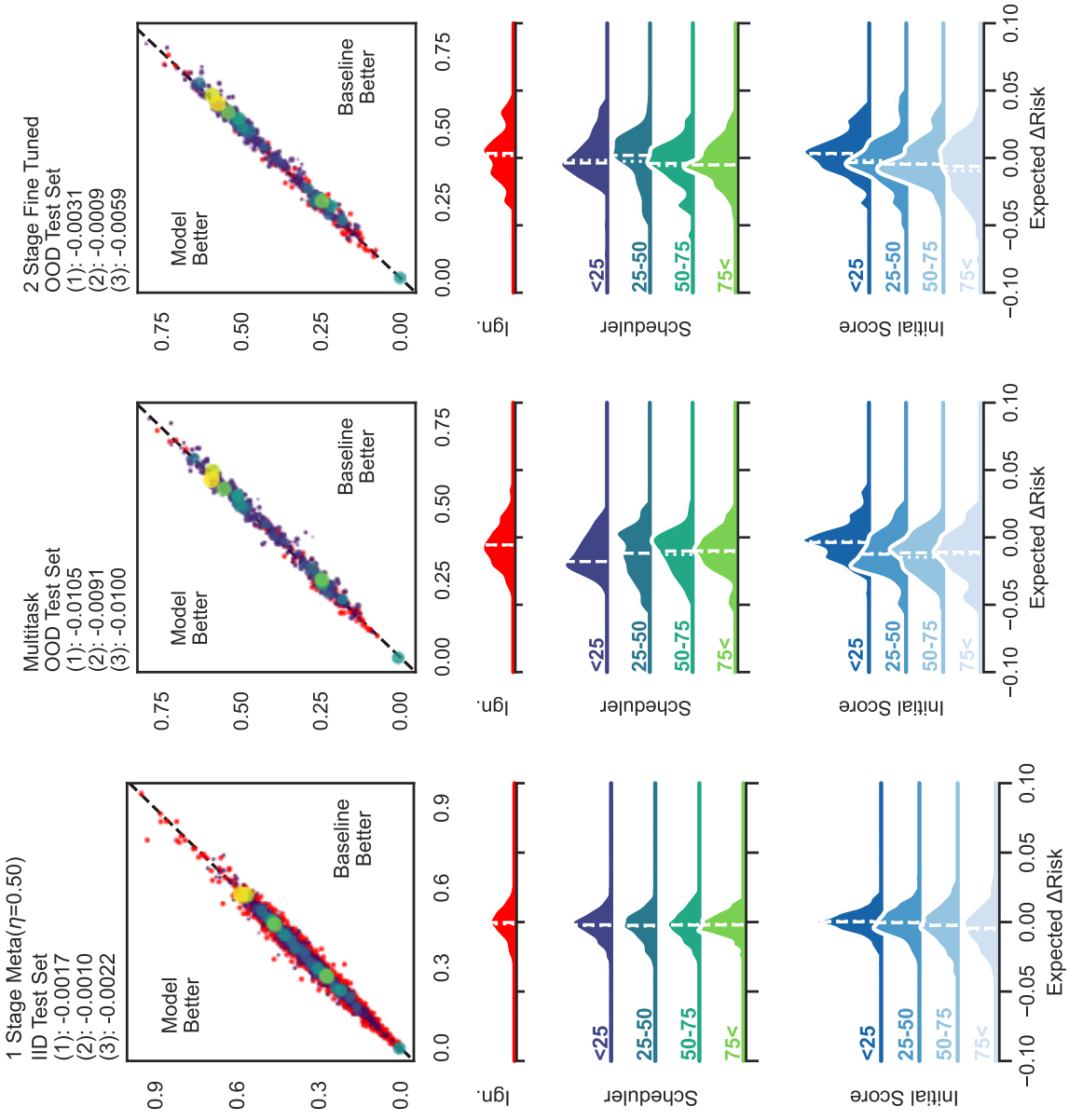


Figure C.4: In the style of Figure 4.8, but now presenting the 2 stage fine-tuned model and the multitask model versus the 1 stage fine-tuned model as baseline. The 2 stage fine-tuned plot for the IID data is missing.

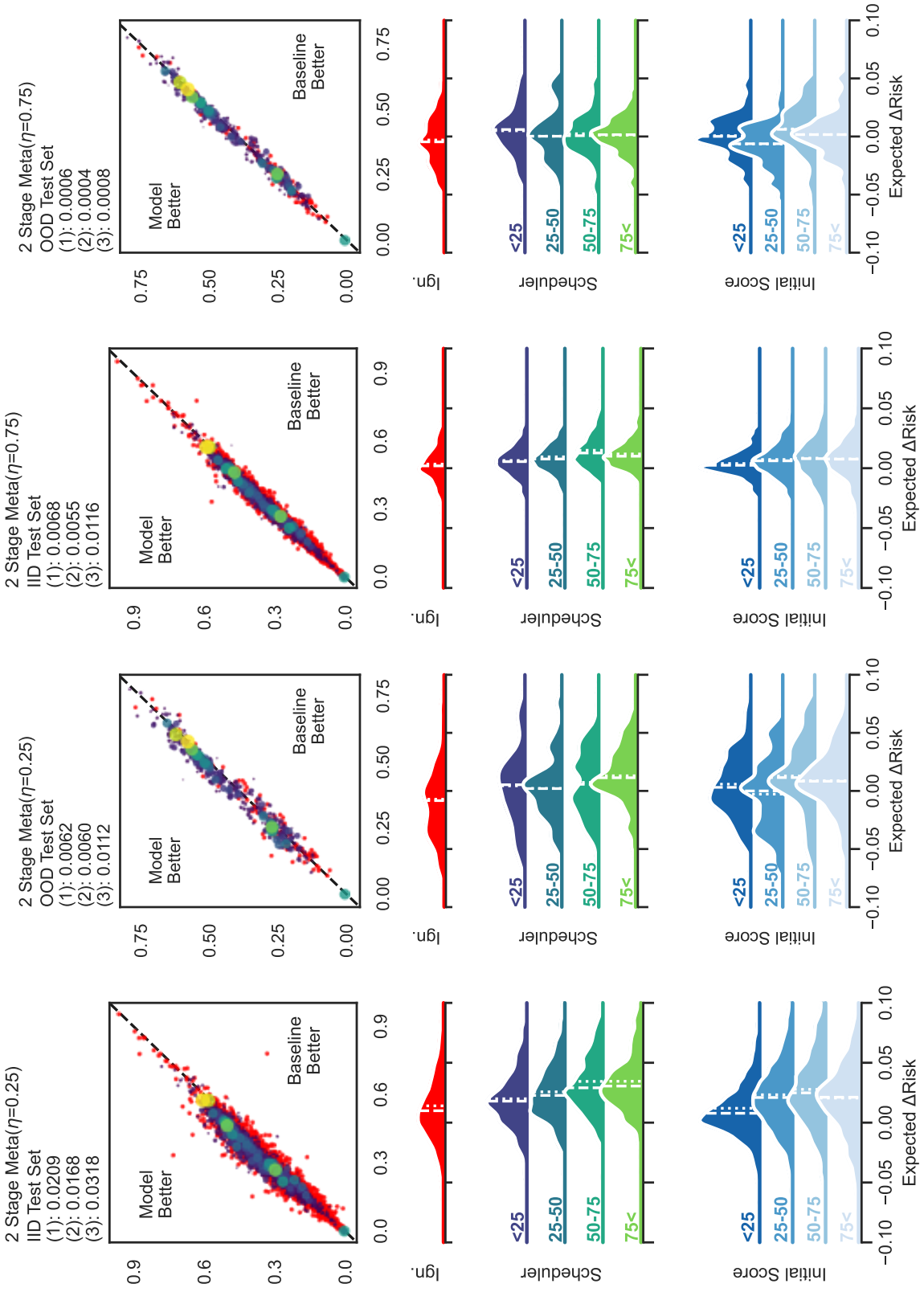


Figure C.5: In the style of Figure 4.8, but now presenting the 2 stage meta-learning adapted model at various values of η versus the 1 stage fine-tuned model as baseline.

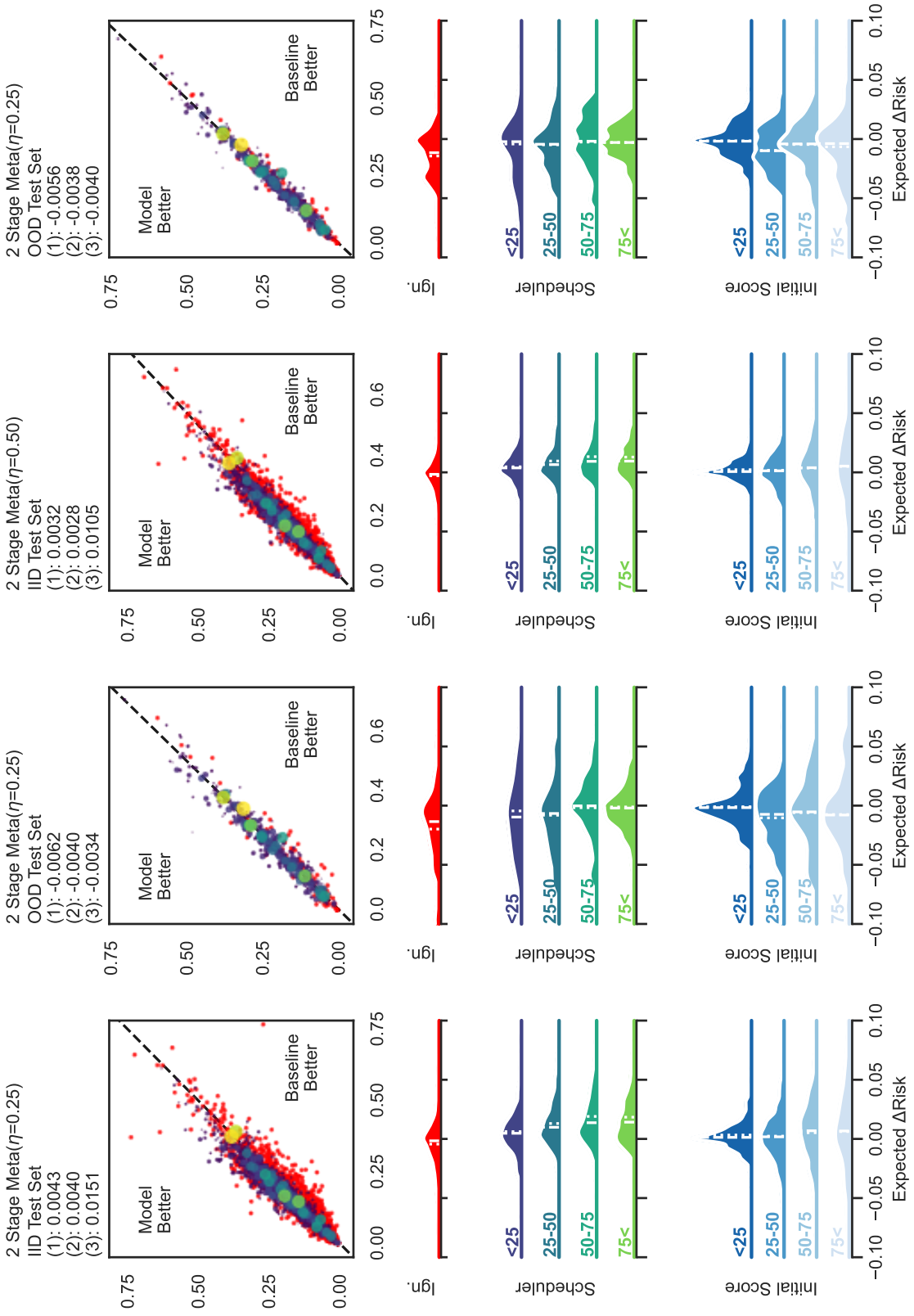


Figure C.6: In the style of Figure 4.8, but now with the character bigram-F1 score, and presenting the 2 stage meta-learning adapted model at various values of η versus the 1 stage fine-tuned model as baseline.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 'Neural machine translation by jointly learning to align and translate'. In: *arXiv preprint arXiv:1409.0473* (2014) (cited on pages 1, 18).

[2] Ashish Vaswani et al. 'Attention is all you need'. In: *Advances in neural information processing systems* 30 (2017) (cited on pages 1, 11, 18).

[3] Jean Berko. 'The child's learning of English morphology'. In: *Word* 14.2-3 (1958), pp. 150–177 (cited on page 1).

[4] Yann LeCun et al. 'Backpropagation applied to handwritten zip code recognition'. In: *Neural computation* 1.4 (1989), pp. 541–551 (cited on page 1).

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 'ImageNet Classification with Deep Convolutional Neural Networks'. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105 (cited on page 1).

[6] Sepp Hochreiter and Jürgen Schmidhuber. 'Long short-term memory'. In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cited on pages 1, 10).

[7] Duygu Ataman, Wilker Aziz, and Alexandra Birch. 'A latent morphology model for open-vocabulary neural machine translation'. In: *arXiv preprint arXiv:1910.13890* (2019) (cited on pages 1, 23, 30, 56).

[8] Martin Haspelmath and Andrea Sims. *Understanding morphology*. Routledge, 2013 (cited on pages 3, 5).

[9] A. Spencer and A.M. Zwicky. *The Handbook of Morphology*. Blackwell Handbooks in Linguistics. Wiley, 1998 (cited on page 3).

[10] Zhou Zhuang. *Zhuangzi*. A translation from Chinese Thought: From Confucius to Cook Ding by Roel Sterckx, page 152. 300BCE (cited on page 4).

[11] Bernard Comrie. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press, 1989 (cited on page 5).

[12] John Sylak-Glassman. 'The composition and use of the universal morphological feature schema (unimorph schema)'. In: *Johns Hopkins University* (2016) (cited on pages 6, 12, 23).

[13] Christo Kirov et al. 'UniMorph 2.0: Universal Morphology'. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018 (cited on pages 6, 12).

[14] Khuyagbaatar Batsuren et al. 'UniMorph 4.0: Universal Morphology'. In: *arXiv preprint arXiv:2205.03608* (2022) (cited on page 6).

[15] Arya D. McCarthy et al. 'The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection'. In: *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 229–244. DOI: 10.18653/v1/W19-4226 (cited on pages 7, 12).

[16] G.A. Chrupala. 'Simple data-driven context-sensitive lemmatization'. English. In: *Procesamiento del Lenguaje natural, Revista* 37 (2006) (cited on page 8).

[17] Eugene W Myers. 'An O(ND) difference algorithm and its variations'. In: *Algorithmica* 1.1 (1986), pp. 251–266 (cited on page 8).

[18] James Coglan. *Building Git*. James Coglan, 2020 (cited on page 8).

[19] Milan Straka, Jana Straková, and Jan Hajič. 'UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging'. In: *arXiv preprint arXiv:1908.06931* (2019) (cited on pages 9, 14).

[20] Milan Straka. 'UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task'. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 197–207. DOI: 10.18653/v1/K18-2020 (cited on page 9).

[21] Piotr Bojanowski et al. 'Enriching Word Vectors with Subword Information'. In: *arXiv preprint arXiv:1607.04606* (2016) (cited on page 9).

[22] Armand Joulin et al. 'Bag of Tricks for Efficient Text Classification'. In: *arXiv preprint arXiv:1607.01759* (2016) (cited on page 9).

[23] Jacob Devlin et al. 'Bert: Pre-training of deep bidirectional transformers for language understanding'. In: *arXiv preprint arXiv:1810.04805* (2018) (cited on page 9).

[24] Yoon Kim et al. 'Character-aware neural language models'. In: *Thirtieth AAAI conference on artificial intelligence*. 2016 (cited on page 10).

[25] Kyunghyun Cho et al. 'On the properties of neural machine translation: Encoder-decoder approaches'. In: *arXiv preprint arXiv:1409.1259* (2014) (cited on page 10).

[26] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. 'Residual LSTM: Design of a deep recurrent architecture for distant speech recognition'. In: *arXiv preprint arXiv:1701.03360* (2017) (cited on page 10).

[27] Milan Straka and Jana Straková. 'UDPipe at EvaLatin 2020: Contextualized embeddings and treebank embeddings'. In: *arXiv preprint arXiv:2006.03687* (2020) (cited on page 10).

[28] Yinhan Liu et al. 'Roberta: A robustly optimized bert pretraining approach'. In: *arXiv preprint arXiv:1907.11692* (2019) (cited on page 10).

[29] Dan Kondratyuk. 'Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning'. In: *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. 2019, pp. 12–18 (cited on pages 10, 14).

[30] Dan Kondratyuk and Milan Straka. '75 languages, 1 model: Parsing universal dependencies universally'. In: *arXiv preprint arXiv:1904.02099* (2019) (cited on page 10).

[31] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 'BERT rediscovers the classical NLP pipeline'. In: *arXiv preprint arXiv:1905.05950* (2019) (cited on page 10).

[32] Jonathan H Clark et al. 'Canine: Pre-training an efficient tokenization-free encoder for language representation'. In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 73–91 (cited on page 11).

[33] Arya D. McCarthy et al. 'Marrying Universal Dependencies and Universal Morphology'. In: *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 91–101. DOI: 10.18653/v1/W18-6011 (cited on pages 11, 12).

[34] Adam Paszke et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035 (cited on page 12).

[35] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014) (cited on page 12).

[36] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/ (cited on pages 13, 39).

[37] Antonios Anastasopoulos. *A note on evaluating multilingual benchmarks*. URL: http://www.cs.cmu.edu/~aanastas/evaluating%5C_multilingual.html (cited on page 13).

[38] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Routledge, 2013 (cited on page 15).

[39] Larry V. Hedges. 'Distribution Theory for Glass's Estimator of Effect Size and Related Estimators'. In: *Journal of Educational Statistics* 6.2 (1981), pp. 107–128. (Visited on 07/01/2022) (cited on page 15).

[40] Chinmay Choudhary. 'Improving the Performance of UDify with Linguistic Typology Knowledge'. In: *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*. Online: Association for Computational Linguistics, June 2021, pp. 38–60. DOI: 10.18653/v1/2021.sigtyp-1.5 (cited on page 17).

[41] Yonatan Belinkov et al. 'What do neural machine translation models learn about morphology?' In: *arXiv preprint arXiv:1704.03471* (2017) (cited on page 18).

[42] Arianna Bisazza and Clara Tump. 'The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation'. In: *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2018, pp. 2871–2876 (cited on pages 18, 20).

[43] Franck Burlot and Francois Yvon. 'Evaluating the morphological competence of Machine Translation Systems'. In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 43–55. DOI: 10.18653/v1/W17-4705 (cited on pages 19, 20, 28).

[44] Franck Burlot et al. 'The WMT'18 Morpheval test suites for English-Czech, English-German, English-Finnish and Turkish-English'. In: *WMT*. 2018 (cited on pages 19, 30).

[45] Yonatan Belinkov and James Glass. 'Analysis methods in neural language processing: A survey'. In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 49–72 (cited on page 19).

[46] Rico Sennrich. 'How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs'. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 376–382 (cited on pages 20, 28).

[47] Adithya Pratapa et al. 'Evaluating the Morphosyntactic Well-formedness of Generated Texts'. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7131–7150. DOI: 10.18653/v1/2021.emnlp-main.570 (cited on pages 20, 21).

[48] Dmytro Kalpakchi and Johan Boye. 'Minor changes make a difference: a case study on the consistency of UD-based dependency parsers'. In: *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*. Sofia, Bulgaria: Association for Computational Linguistics, Dec. 2021, pp. 96–108 (cited on page 20).

[49] Jörg Tiedemann and Santhosh Thottingal. 'OPUS-MT — Building open translation services for the World'. In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal, 2020 (cited on page 22).

[50] Marcin Junczys-Dowmunt et al. 'Marian: Fast Neural Machine Translation in C++'. In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia, 2018 (cited on page 22).

[51] Rico Sennrich, Barry Haddow, and Alexandra Birch. 'Neural Machine Translation of Rare Words with Subword Units'. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162 (cited on pages 22, 30).

[52] Mike Lewis et al. 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension'. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703 (cited on page 22).

[53] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. DOI: 10.48550/ARXIV.1910.03771. URL: https://arxiv.org/abs/1910.03771 (cited on page 22).

[54] Jörg Tiedemann. 'Parallel Data, Tools and Interfaces in OPUS'. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey: European Language Resources Association (ELRA), 2012 (cited on page 22).

[55] Jörg Tiedemann. 'The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT'. In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1174–1182 (cited on page 22).

[56] Ari Holtzman et al. 'The curious case of neural text degeneration'. In: *arXiv preprint arXiv:1904.09751* (2019) (cited on page 23).

[57] JASP Team. *JASP (Version 0.16.3)[Computer software]*. 2022. URL: https://jasp-stats.org/ (cited on page 24).

[58] Taku Kudo and John Richardson. 'SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing'. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012 (cited on page 30).

[59] Sabrina J Mielke et al. 'Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP'. In: *arXiv preprint arXiv:2112.10508* (2021) (cited on page 30).

[60] Duygu Ataman et al. 'On the Importance of Word Boundaries in Character-level Neural Machine Translation'. In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 187–193. DOI: 10.18653/v1/D19-5619 (cited on pages 30, 31).

[61] Duygu Ataman et al. 'Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English.' In: (2017) (cited on page 30).

[62] Duygu Ataman and Marcello 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)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 305–311. DOI: 10.18653/v1/P18-2049 (cited on page 30).

[63] Diederik P Kingma and Max Welling. 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114* (2013) (cited on page 31).

[64] Steven Y Feng et al. 'A survey of data augmentation approaches for nlp'. In: *arXiv preprint arXiv:2105.03075* (2021) (cited on page 31).

[65] Chris Hokamp and Qun Liu. 'Lexically constrained decoding for sequence generation using grid beam search'. In: *arXiv preprint arXiv:1704.07138* (2017) (cited on page 31).

[66] Matt Post and David Vilar. 'Fast lexically constrained decoding with dynamic beam allocation for neural machine translation'. In: *arXiv preprint arXiv:1804.06609* (2018) (cited on page 31).

[67] Maria Nadejde et al. 'Predicting target language CCG supertags improves neural machine translation'. In: *arXiv preprint arXiv:1702.01147* (2017) (cited on page 31).

[68] Aleš Tamchyna, Marion Weller-Di Marco, and Alexander Fraser. 'Modeling target-side inflection in neural machine translation'. In: *arXiv preprint arXiv:1707.06012* (2017) (cited on pages 31, 32).

[69] Costanza Conforti, Matthias Huck, and Alexander Fraser. 'Neural morphological tagging of lemma sequences for machine translation'. In: *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. 2018, pp. 39–53 (cited on pages 31, 32).

[70] Marion Weller-Di Marco, Matthias Huck, and Alexander Fraser. 'Modeling Target-Side Morphology in Neural Machine Translation: A Comparison of Strategies'. In: *arXiv preprint arXiv:2203.13550* (2022) (cited on pages 31, 32).

[71] Fahim Dalvi et al. 'Understanding and improving morphological learning in the neural machine translation decoder'. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2017, pp. 142–151 (cited on page 32).

[72] Georgiana Dinu et al. 'Training Neural Machine Translation to Apply Terminology Constraints'. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3063–3068. DOI: 10.18653/v1/P19-1294 (cited on page 32).

[73] Miriam Exel et al. 'Terminology-Constrained Neural Machine Translation at SAP'. In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Lisboa, Portugal: European Association for Machine Translation, Nov. 2020, pp. 271–280 (cited on pages 32, 33).

[74] Toms Bergmanis and Mārcis Pinnis. 'Facilitating terminology translation with target lemma annotations'. In: *arXiv preprint arXiv:2101.10035* (2021) (cited on pages 32, 33).

[75] Josef Jon et al. 'End-to-End Lexically Constrained Machine Translation for Morphologically Rich Languages'. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4019–4033. DOI: 10.18653/v1/2021.acl-long.311 (cited on page 33).

[76] Weijia Xu and Marine Carpuat. 'Rule-based Morphological Inflection Improves Neural Terminology Translation'. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5902–5914. DOI: 10.18653/v1/2021.emnlp-main.477 (cited on page 33).

[77] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 'Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks'. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1126–1135 (cited on page 34).

[78] Jiatao Gu et al. 'Meta-learning for low-resource neural machine translation'. In: *arXiv preprint arXiv:1808.08437* (2018) (cited on page 34).

[79] Nier Wu et al. 'Low-Resource Neural Machine Translation Based on Improved Reptile Meta-learning Method'. In: *China Conference on Machine Translation*. Springer. 2021, pp. 39–50 (cited on page 34).

[80] Alex Nichol, Joshua Achiam, and John Schulman. 'On First-Order Meta-Learning Algorithms'. In: *CoRR* abs/1803.02999 (2018) (cited on page 35).

[81] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. 'On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms'. In: *CoRR* abs/1908.10400 (2019) (cited on page 35).

[82] Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. 'How to train your MAML'. In: *CoRR* abs/1810.09502 (2018) (cited on page 35).

[83] Aniruddh Raghu et al. 'Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML'. In: *International Conference on Learning Representations*. 2020 (cited on pages 35, 38, 54, 55).

[84] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. 'How Does the Task Landscape Affect MAML Performance?' In: *arXiv preprint arXiv:2010.14672* (2020) (cited on page 36).

[85] Sébastien Arnold et al. 'Uniform Sampling over Episode Difficulty'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1481–1493 (cited on page 36).

[86] Chenghao Liu et al. 'Adaptive task sampling for meta-learning'. In: *European Conference on Computer Vision*. Springer. 2020, pp. 752–769 (cited on page 36).

[87] Marta R Costa-jussà et al. 'No Language Left Behind: Scaling Human-Centered Machine Translation'. In: *arXiv e-prints* (2022), arXiv–2207 (cited on page 37).

[88] Antreas Antoniou, Harrison Edwards, and Amos Storkey. 'How to train your MAML'. In: *arXiv preprint arXiv:1810.09502* (2018) (cited on page 38).

[89] NLLB Team et al. 'No Language Left Behind: Scaling Human-Centered Machine Translation'. In: () (cited on page 44).

[90] Sébastien Arnold, Shariq Iqbal, and Fei Sha. 'When maml can adapt fast and how to assist when it cannot'. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 244–252 (cited on pages 54, 55).

[91] Jaehoon Oh et al. '{BOIL}: Towards Representation Change for Few-shot Learning'. In: *International Conference on Learning Representations*. 2021 (cited on page 55).

[92] Johannes Von Oswald et al. 'Learning where to learn: Gradient sparsity in meta and continual learning'. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. 2021 (cited on page 55).

[93] Kishore Papineni et al. 'Bleu: a method for automatic evaluation of machine translation'. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318 (cited on page 55).

[94] Tom Kocmi et al. 'To ship or not to ship: An extensive evaluation of automatic metrics for machine translation.' In: *Proceedings of the 6th Conference on Machine Translation of the Association for Computational Linguistics*. 2021, pp. 478–494 (cited on pages 55, 56).

[95] Matt Post. 'A Call for Clarity in Reporting BLEU Scores'. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191 (cited on page 56).

[96] Maja Popović. 'chrF++: words helping character n-grams'. In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 612–618. DOI: 10.18653/v1/W17-4770 (cited on page 56).

[97] Maja Popović. 'chrF: character n-gram F-score for automatic MT evaluation'. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 392–395. DOI: 10.18653/v1/W15-3049 (cited on page 56).

[98] Ricardo Rei et al. 'COMET: A Neural Framework for MT Evaluation'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2685–2702. DOI: 10.18653/v1/2020.emnlp-main.213 (cited on page 56).

[99] Nitika Mathur et al. 'Results of the WMT20 Metrics Shared Task'. In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 688–725 (cited on page 56).

[100] Markus Freitag et al. 'Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain'. In: *Proceedings of the Sixth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2021, pp. 733–774 (cited on page 56).

[101] Markus Freitag et al. 'Experts, errors, and context: A large-scale study of human evaluation for machine translation'. In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1460–1474 (cited on page 56).