Morphological Analysis Using a Sequence Decoder

Ekin Akyürek* **Erenay Dayanık*** **Deniz Yuret†**

Koç University Artificial Intelligence Laboratory, İstanbul, Turkey

eakyurek13,edayanik16,dyuret@ku.edu.tr

Abstract

We introduce Morse, a recurrent encoder-decoder model that produces morphological analyses of each word in a sentence. The encoder turns the relevant information about the word and its context into a fixed size vector representation and the decoder generates the sequence of characters for the lemma followed by a sequence of individual morphological features. We show that generating morphological features individually rather than as a combined tag allows the model to handle rare or unseen tags and to outperform whole-tag models. In addition, generating morphological features as a sequence rather than, for example, an unordered set allows our model to produce an arbitrary number of features that represent multiple inflectional groups in morphologically complex languages. We obtain state-of-the-art results in nine languages of different morphological complexity under low-resource, high-resource, and transfer learning settings. We also introduce TrMor2018, a new high-accuracy Turkish morphology data set. Our Morse implementation and the TrMor2018 data set are available online to support future research.[1]

1 Introduction

Morse is a recurrent encoder-decoder model that takes sentences in plain text as input and produces both lemmas and morphological features of each word as output. Table 1 presents an example: The ambiguous Turkish word ‘‘*masalı*’’ has three possible morphological analyses: the accusative and possessive forms of the stem ‘‘*masal*’’ (tale) and the `+With` form of the stem ‘‘*masa*’’ (table), all expressed with the same surface form (Oflazer, 1994). Morse attempts to output the correct analysis of each word based on its context in a sentence.

Accurate morphological analysis and disambiguation are important prerequisites for further syntactic and semantic processing, especially in morphologically complex languages. Many languages mark case, number, person, and so on. using morphology, which helps discover the correct syntactic dependencies. In agglutinative languages, syntactic dependencies can even be between subword units. For example, Oflazer et al. (1999) observes that words in Turkish can have dependencies to any one of the inflectional groups of a derived word: in ‘‘*mavi masalı oda*’’ (room with a blue table) the adjective ‘‘*mavi*’’ (blue) modifies the noun root ‘‘*masa*’’ (table) even though the final part of speech of ‘‘masalı’’ is an adjective. This dependency would be difficult to represent without a detailed analysis of morphology.

We combined the following ideas to attack morphological analysis in the Morse model:

- Morse does not require an external rule-based analyzer or dictionary, avoiding the parallel maintenance of multiple systems.
- Morse performs lemmatization and tagging jointly by default; we also report on separating the two tasks.
- Morse outputs morphological tags one feature at a time, giving it the ability to learn unseen/rare tags.
- Morse generates features as a variable size seqeunce rather than a fixed set, allowing it to represent derivational morphology.

*Equal contribution.

†Corresponding author.

[1]See `https://github.com/ai-ku/Morse.jl` for a Morse implementation in Julia/Knet (Yuret, 2016) and `https://github.com/ai-ku/TrMor2018` for the new Turkish data set.

Transactions of the Association for Computational Linguistics, vol. 7, pp. 567–579, 2019. https://doi.org/10.1162/tacl_a_00286
Action Editor: Hinrich Schütze. Submission batch: 12/2018; Revision batch: 4/2019; Published 9/2019.

Context & analysis of ‘‘masalı’’
masalı yaz. (write **the tale**.) masal+Noun+A3sg+Pnon+Acc
babamın **masalı** (my father’s **tale**) masal+Noun+A3sg+P3sg+Nom
mavi **masalı** oda (room **with a** blue **table**) masa+Noun+A3sg+Pnon+NomˆDB+Adj+With

Table 1: Morphological analyses for Turkish word *masalı*. An example context and its translation is given before each analysis.

We evaluated our model on several Turkish data sets (Yuret and Türe, 2006; Yıldız et al., 2016) and eight languages from the Universal Dependencies data set (UD; Nivre et al., 2016) in low-resource, high-resource, and transfer learning settings for comparison with existing work. We realized that existing Turkish data sets either had low inter-annotator agreement or small test sets, which made model comparison difficult because of noise and statistical significance problems. To address these issues we also created a new Turkish data set, TrMor2018, which contains 460 K tagged tokens and has been verified to be 96% accurate by trained annotators. We report our results on this new data set as well as previously available data sets.

The main contributions of this work are:

- A new encoder-decoder model that performs joint lemmatization and morphological tagging which can handle unknown words, unseen tag sequences, and multiple inflectional groups.
- State-of-the-art results on nine languages of varying morphological complexity in low-resource, high-resource, and transfer learning settings.
- Release of a new morphology data set for Turkish.

We discuss related work in Section 2, detail our model’s input output representation and individual components in Section 3, describe our data sets and introduce our new Turkish data set in Section 4, present our experiments and results in Section 5, and conclude in Section 6.

2 Related Work

Morphological word analysis has been typically performed by solving multiple subproblems. In one common approach the subproblems of *lemmatization* (e.g., finding the stem ‘‘masal’’ for the first two examples in Table 1 and ‘‘masa’’ for the third) and *morphological tagging* (e.g., producing `+Noun+A3sg+Pnon+Acc` in the first example) are attacked separately. In another common approach a language-dependent rule-based *morphological analyzer* outputs all possible lemma+tag analyses for a given word, and a statistical *disambiguator* picks the correct one in a given context. Even though Morse attacks these problems jointly, the prior work is best presented within these traditional divisions, contrasting various approaches with Morse where appropriate.

2.1 Lemmatization and Tagging

Early work in this area typically performed lemmatization and tagging separately. For example, the Shortest Edit Script (SES) approach to lemmatization classifies lemmas based on the mimimum sequence of operations that converts a wordform into a lemma (Chrupala, 2006). MarMoT (Mueller et al., 2013) predicts the sequence of morphological tags in a sentence using a pruned higher-order conditional random field.

SES was later extended to do joint lemmatization and morphological tagging in Morfette (Chrupala et al., 2008), where two separate maximum entropy models are trained for predicting the lemma and the morphological tag and a third model returns a probability distribution over lemma-tag pairs. MarMoT was extended to Lemming (Müller et al., 2015), which used a joint log-linear model of lemmatization and tagging and provided empirical evidence that jointly modeling morphological tags and lemmata is mutually beneficial.

We chose to perform lemmatization and tagging jointly in Morse partly for linguistic reasons: as Table 1 shows, a tag like `+Noun+A3sg+Pnon+Acc` can be correct with respect to one lemma (`masal`) and not another (`masa`). For comparison with some of the earlier work, we did train Morse to only generate the morphological tag and observed some improvement in low-resource and transfer-learning settings, but no significant improvement in high-resource experiments.

More recent work started experimenting with deep learning models. Heigold et al. (2017) outperformed MarMoT in morphological tagging

Language	100sent	1000sent
Swedish	9.19	1.02
Bulgarian	14.38	2.68
Hungarian	15.78	3.93
Portuguese	6.04	0.82

Table 2: Percentage of tags in the test data that have been observed fewer than 5 times in the training data for four languages and two training sizes (100 and 1000 sentences).

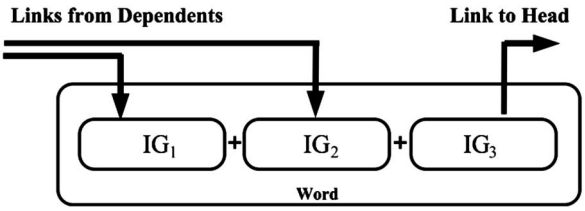

Figure 1: Multiple inflectional groups in a word may have independent syntactic relationships. Figure from Eryiğit and Oflazer (2006).

using a character-based recurrent neural network encoder similar to Morse, combined with a whole-tag classifier. To address the data sparseness problem this work was extended in Cotterell and Heigold (2017) with transfer learning, improving performance on low resource languages by up to 30% using a related-high resource language.

Morse uses a character-based encoder that turns the relevant features of the word and its context into fixed-size vector representations similar to Heigold et al. (2017). Our main contribution is the *sequence decoder* that generates the characters of the lemma and/or morphological features sequentially one at a time. This is similar to the way rule-based systems such as finite state transducers output morphological analyses. One advantage of generating features one at a time (e.g., `+Acc`) rather than as a combined tag (e.g., `+Noun+A3sg+Pnon+Acc`) is sample efficiency. Table 2 shows the percentage of tags in the test data that have been observed rarely in the training data for several languages. In low resource experiments, we show that our sequence decoder significantly outperforms a variant that is trained to output full tags similar to Heigold et al. (2017), especially with unseen or rare tags.

Malaviya et al. (2018) also avoid the data sparsity problem associated with whole tags using a neural factor graph model to predict a set of features, improving the transfer learning performance. In contrast with Malaviya et al. (2018), Morse generates a variable number of features as a sequence rather than a fixed set. This allows it to adequately represent derivations in morphologically complex words. For example, in the last analysis in Table 1, morphological features of the word ‘‘*masalı*’’ consist of two inflectional groups (IGs), a noun group and an adjective group, separated by a derivational boundary denoted by ‘‘`^DB`’’. In ‘‘*mavi masalı oda*’’ (room with a blue table) the adjective ‘‘*mavi*’’ (blue) modifies the noun root ‘‘*masa*’’ (table) even though the final part of speech of ‘‘*masalı*’’ is an adjective. In general, each IG in a morphologically complex word may have independent syntactic dependencies, as shown in Figure 1. Thus, for languages like Turkish, it is linguistically essential to be able to represent multiple IGs with a variable number of features (Eryiğit et al., 2008). The sequence-decoder approach of Morse outperforms the neural factor graph model of Malaviya et al. (2018) in both low-resource and transfer learning settings.

2.2 Analysis and Disambiguation

Morphological analysis is the task of producing all possible morphological parses for a given word. For morphologically simple languages like English, a dictionary is typically sufficient for this task (Baayen et al., 1995). For morphologically complex languages like Turkish, the analysis can be performed by language dependent rule-based systems such as finite-state transducers that encode morphophonemics and morphotactics (Koskenniemi, 1981, 1983; Karttunen and Wittenburg, 1983). The first rule-based analyzer for Turkish was developed in Oflazer (1994), we used an updated version of this analyzer (Oflazer, 2018) when creating our new Turkish data set.

Morphological disambiguation systems take the possible parses for a given word from an analyzer and predict the correct one in a given context using rule-based (Karlsson et al., 1995; Oflazer and Kuruöz, 1994; Oflazer and Tür, 1996; Daybelge and Çiçekli, 2007; Daoud, 2009), statistical (Hakkani-Tür et al., 2002; Yuret and Türe, 2006; Hajič et al., 2007), or neural network based (Yıldız et al., 2016; Shen et al., 2016; Toleu et al., 2017) techniques. Hakkani-Tür et al. (2018) provide a comprehensive summary for Turkish disambiguators.

Morse performs morphological analysis and disambiguation with a joint model partly to avoid using a separate morphological analyzer or dictionary. Having a single system combining morphological analysis and disambiguation is easier to use and maintain. The additional constraints brought by an external morphological analyzer or dictionary are certainly beneficial, but the benefit appears to be limited with sufficient data: In our experiments, (1) we outperform earlier systems that use separate morphological analysis and disambiguation components, and (2) when we use Morse only to disambiguate among the analyses generated by a rule-based analyzer, the accuracy gain is less than 1% compared with generating analyses from scratch.

3 Model

Morse produces the morphological analysis (lemma plus morphological features) for each word in a given sentence. It is loosely based on the sequence-to-sequence encoder-decoder network approach proposed by Sutskever et al. (2014) for machine translation. However, we use three distinct encoders to create embeddings of various input features. First, a word encoder creates an embedding for each word based on its characters. Second, a context encoder creates an embedding for the context of each word based on the word embeddings of all words to the left and to the right. Third, an output encoder creates an output embedding using the morphological features of the last two words. These embeddings are fed to the decoder, which produces the lemma and the morphological features of a target word one character/feature at a time. In the following subsections, we explain each component in detail.

3.1 Input Output

The input to the model consists of an N word sentence $S = [w_1, \ldots, w_N]$, where w_i is the i'th word in the sentence. Each word is input as a sequence of characters $w_i = [w_{i1}, \ldots, w_{iL_i}], w_{ij} \in \mathcal{A}$ where $\mathcal{A}$ is the set of alphanumeric characters and L_i is the number of characters in word w_i.

The output for each word consists of a lemma, a part-of-speech tag and a set of morphological features—for example, [m, a, s, a, l, Noun, A3sg, P3sg, Nom] for ‘‘masalı’’. The lemma is produced one character at a time, and the morphological information is produced one feature at a time. A sample output for a word looks like $[s_{i1}, \ldots, s_{iR_i}, f_{i1}, \ldots, f_{iM_i}]$ where $s_{ij} \in \mathcal{A}$ is an alphanumeric character in the lemma, R_i is the length of the lemma, M_i is the number of features, and $f_{ij} \in \mathcal{T}$ is a morphological feature from a feature set such as $\mathcal{T} = \{\text{Noun,Adj,Nom,A3sg,} \ldots\}$.

We have experimented with other input-output formats, as described in Section 5: We found that jointly producing the lemma and the morphological features is more difficult than producing only morphological features in low-resource settings but gives similar performance in high-resource settings. We also found that generating the morphological tag one feature at a time rather than as a complete tag is advantageous, more so in morphologically complex languages and in low-resource settings.

3.2 Word Encoder

We map each character w_{ij} to an A dimensional character embedding vector $a_{ij} \in \mathbb{R}^A$. The word encoder takes each word and processes the character embeddings from left to right producing hidden states $[h_{i1}, \ldots, h_{iL_i}]$ where $h_{ij} \in \mathbb{R}^H$. The final hidden state $e_i = h_{iL_i}$ is used as the word embedding for word w_i. The top left box in Figure 2 depicts the word encoder. We also experimented with external word embeddings but did not observe any significant improvement.

$$h_{ij} = \text{LSTM}(a_{ij}, h_{ij-1}) \quad (1)$$

$$h_{i0} = 0 \quad (2)$$

$$e_i = h_{iL_i} \quad (3)$$

3.3 Context Encoder

We use a bidirectional long short-term memory network (LSTM) for the context encoder. The inputs are the word embeddings $e_1, \cdots, e_N$ produced by the word encoder. The context encoder processes them in both directions and constructs a unique context embedding for each target word in the sentence. For a word w_i we define its corresponding context embedding $c_i \in \mathbb{R}^{2H}$ as the concatenation of the forward $\vec{c}_i \in \mathbb{R}^H$ and the backward $\overleftarrow{c}_i \in \mathbb{R}^H$ hidden states that are produced after the forward and backward LSTMs process the word embedding e_i.

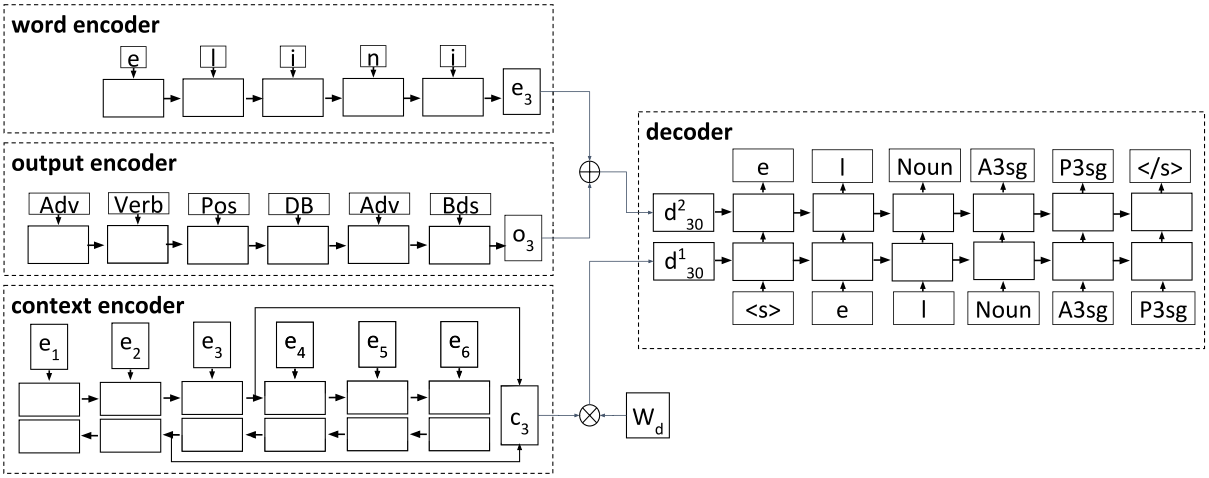

Figure 2: Model illustration for the sentence *"Sonra gülerek elini kardeşinin omzuna koydu"* (Then he laughed and put his hand on his brother's shoulder) and target word *''elini''* (his hand). We use the morphological features of the words preceding the target as input to the output encoder: ''Sonra+Adv gül+Verb+Posˆ DB+Adverb+ByDoingSo''.

The bottom left box in Figure 2 depicts creation of the context vector for the target word ''elini''.

$$\vec{c}_i = \text{LSTM}_f(e_i, \vec{c}_{i-1}) \quad (4)$$

$$\overleftarrow{c}_i = \text{LSTM}_b(e_i, \overleftarrow{c}_{i+1}) \quad (5)$$

$$\vec{c}_0 = \overleftarrow{c}_{N+1} = 0 \quad (6)$$

$$c_i = [\vec{c}_i; \overleftarrow{c}_i] \quad (7)$$

3.4 Output Encoder

The output encoder captures information about the morphological features of words processed prior to each target word. For example, in order to assign the correct possessive marker to the word ''*masalı*'' (tale) in ''*babamın masalı*'' (my father's tale), it would be useful to know that the previous word ''*babamın*'' (my father's) has a genitive marker. During training we use the gold morphological features, during testing we use the output of the model.

The output encoder only uses the morphological features, not the lemma characters, of the previous words as input: $[f_{11}, \ldots, f_{1M_1}, f_{21}, \ldots, f_{i-1,M_{i-1}}]$. We map each morphological feature f_{ij} to a B dimensional feature embedding vector $b_{ij} \in \mathbb{R}^B$. A unidirectional LSTM is run over the morphological features of the last two words to produce hidden states $[t_{11}, \ldots, t_{i-1,M_{i-1}}]$ where $t_{ij} \in \mathbb{R}^H$. The final hidden state preceding the target word $o_i = t_{i-1,M_{i-1}}$ is used as the output embedding for word w_i. The middle left box in Figure 2 depicts the output encoder.

$$t_{ij} = \text{LSTM}(b_{ij}, t_{ij-1}) \quad (8)$$

$$t_{i0} = t_{i-1,M_{i-1}} \quad (9)$$

$$o_i = t_{i-1,M_{i-1}} \quad (10)$$

3.5 Decoder

The decoder is implemented as a 2-layer LSTM network that outputs the correct lemma+tag for a single target word.[2] By conditioning on the three encoder embeddings and its own hidden state, the decoder learns to generate $y_i = [y_{i1}, \ldots, y_{iK_i}]$ where y_i is the correct sequence for the target word w_i in sentence S, $y_{ij} \in \mathcal{A} \cup \mathcal{T}$ represents both lemma characters and morphological feature tokens, and K_i is the total number of output tokens (lemma + features) for word w_i. The first layer of the decoder is initialized with the context embedding c_i.

$$d^1_{i0} = \text{relu}(W_d \times c_i \oplus W_{db}) \quad (11)$$

$$d^1_{ij} = \text{LSTM}(y_{ij-1}, d^1_{ij-1}) \quad (12)$$

where $W_d \in \mathbb{R}^{H \times 2H}$, $W_{db} \in \mathbb{R}^H$, and $\oplus$ is element-wise summation. We initialize the second

[2]We also experimented with two variants of our model: MorseTag only outputs morphological features, and MorseDisamb uses the decoder to rank probabilities of a set of analyses provided by a rule-based system.

lang	train	dev	test	|T|	|F|	|R|	lang	train	dev	test	|T|	|F|	|R|
DA	80378	10332	10023	159	44	0.03%	SV	66645	9797	20377	211	40	0.06%
RU	75964	11877	11548	734	39	0.27%	BG	124336	16089	15724	439	45	0.03%
FI	162621	18290	21041	2243	93	0.68%	HU	20166	11418	10448	716	73	1.03%
ES	384554	37349	12069	404	46	0.03%	PT	211820	11158	10468	380	47	0.03%

Table 3: Data statistics of UD Version 2.1 Treebanks. The values in the {train, dev, test} columns are the number of tokens in the splits. $|T|$ gives the number of distinct tags (pos + morphological features), $|F|$ the number of distinct feature values. |R| gives the unseen tag percentage in the test set.

layer with the word and output embeddings after combining them by element-wise summation.

$$d^2_{i0} = e_i + o_i \quad (13)$$

$$d^2_{ij} = \text{LSTM}(d^1_{ij}, d^2_{ij-1}) \quad (14)$$

We parameterize the distribution over possible morphological features and characters at each time step as

$$p(y_{ij}|d^2_{ij}) = \text{softmax}(W_s \times d^2_{ij} \oplus W_{sb}) \quad (15)$$

where, $W_s \in \mathbb{R}^{|\mathcal{Y}|\times H}$, and $W_{sb} \in \mathbb{R}^{|\mathcal{Y}|}$ where $\mathcal{Y} = \mathcal{A} \cup \mathcal{T}$ is the set of characters and morphological features in output vocabulary. The right side of Figure 2 depicts the decoder.

4 Data Sets

We evaluate Morse on several different languages and data sets. First we describe the multilingual data sets we used from the UD data sets (Nivre et al., 2016). We then describe two existing data sets for Turkish and introduce our new data set TrMor2018.

4.1 Universal Dependency Data Sets

We tested Morse on eight languages selected from the UD data sets Version 2.1 (Nivre et al., 2016). In Table 3, we summarize the corpus statistics. Specifically, we use the CoNLL-U format[3] for the input files, take column 2 (FORM) as input, and predict columns 3 (LEMMA), 4 (UPOSTAG), and 6 (FEATS). We show the number of distinct features with $|F|$, the number of distinct composite tags with $|T|$, and the unseen composite tag percentage with $|R|$ to indicate the morphological complexity of a language.

4.2 Turkish Data Sets

For Turkish we evaluate our model on three data sets described in Table 4. These data sets contain derivational as well as inflectional morphology represented by multiple inflectional groups as described in the Introduction. In contrast, the UD data sets only preserve information in the last inflectional group.

Dataset	Ambig	Unamb	Total
TrMor2006Train	398290	439234	837524
TrMor2006Test	379	483	862
TrMor2016Test	9460	9802	19262
TrMor2018	216803	243866	460669

Table 4: Number of ambiguous, unambiguous, and all tokens for data sets TrMor2006 (Yuret and Türe, 2006), TrMor2016 (Yıldız et al., 2016) (which shares the same training set), and TrMor2018 (introduced in this paper).

The first data set, TrMor2006, was provided by Kemal Oflazer and published in Yuret and Türe (2006) based on a Turkish newspaper data set. The training set was disambiguated semi-automatically and has limited accuracy. The test set was hand-tagged but is very small (862 tokens) to reliably distinguish between models with similar accuracy. We randomly extracted 100 sentences from the training set and used them as the development set while training our model.

The second data set, TrMor2016, was prepared by Yıldız et al. (2016). The training set is the same as TrMor2006 but they manually retagged a subset of the training set containing roughly 20,000 tokens to be used as a larger test set. Unfortunately they did not exclude the sentences in the test set from the training set in their experiments. Furthermore, they do not provide any inter-annotator agreement results on the new test set.

Given the problems associated with these data sets, we decided to prepare a new data set, TrMor2018, that we release with this paper. Our goal is to provide a data set with high inter-annotator agreement that is large enough to allow

[3] http://universaldependencies.org/format.html

dev/test sets of sufficient size to distinguish model performances in a statistically significant manner. The new data set consists of 34,673 sentences and 460,669 tokens in total from different genres of Turkish text.

TrMor2018 was annotated semi-automatically in multiple passes. The initial pass was performed automatically by a previous state-of-the-art model (Yuret and Türe, 2006). The resulting data were spot checked in multiple passes for mistakes and inconsistencies by annotators, prioritizing ambiguous high-frequency words. Any systematic errors discovered were corrected by hand-written scripts.

In order to monitor our progress, we randomly selected a subset and disambiguated all of it manually. This subset contains 2,090 sentences and 26,819 words. Two annotators annotated each word independently and we assigned the final morphological tag of each word based on the adjudication by a third. Taking this hand-tagged subset as the gold standard, we measure the noise level in the corresponding semi-automatic results after every pass. Importantly, the hand-tagged subset is only used for evaluating the noise level of the main data set (i.e., we do not use it for training or testing, and we do not use the identity of the mistakes to inform our passes). Our current release of TrMor2018 has a disagreement level of 4.4% with the hand-tagged subset, which is the current state-of-the-art for Turkish morphological data sets.

5 Experiments and Results

In this section we describe our training procedure, give experimental results, compare with related models, and provide an ablation analysis. The results demonstrate that Morse, generating analyses with its sequence decoder, significantly outperforms the state of the art in low-resource, high-resource, and transfer-learning experiments. We also experimented with two variants of our model for more direct comparisons: MorseTag which only predicts tags without lemmas, and MorseDisamb which chooses among the analyses generated by a rule-based morphological analyzer.

5.1 Training

All LSTM units have $H = 512$ hidden units in our experiments. The size of the character embedding vectors are $A = 64$ in the word encoder. In the decoder part, the size of the output embedding vectors is $B = 256$. We initialized model parameters with Xavier initialization (Glorot and Bengio, 2010).

Our networks are trained using back-propagation through time with stochastic gradient descent. The learning rate is set to $lr = 1.6$ and is decayed based on the development accuracy. We apply learning rate decay by a factor of 0.8 if the development set accuracy is not improved after 5 consecutive epochs. Likewise, early-stopping is forced if the development set accuracy is not improved after 10 consecutive epochs, returning the model with the best dev accuracy. To reduce overfitting, dropout is applied with the rates of 0.5 for low-resource and 0.3 in high-resource settings for each of the LSTM units as well as embedding layers.

5.2 Multilingual Results

For comparison with existing work, we evaluated our model on four pairs of high/low resource language pairs: Danish/Swedish (DA/SV), Russian/Bulgarian (RU/BG), Finnish/Hungarian (FI/HU), and Spanish/Portuguese (ES/PT). Table 5 compares the accuracy and Table 6 compares the F1 scores of four related models:[4] (1) Cotterell: a classification-based model with a similar encoder that predicts whole tags rather than individual features (Cotterell and Heigold, 2017), (2) Malaviya: a neural factor graph model that predicts a fixed number of morphological features rather than variable length feature sequences (Malaviya et al., 2018), (3) Morse: our model with joint prediction of the lemma and the tag (the lemma is ignored in scoring), and (4) MorseTag: a version of our model that predicts only the morphological tag without the lemma (Cotterell and Malaviya only predict tags). We compare results in three different settings: (1) LR100 and LR1000 columns show the low-resource setting where we experiment with 100 and 1000 sentences of training data in Swedish, Bulgarian, Hungarian, and Portuguese, (2) XFER100 and XFER1000 columns show the transfer learning setting where the related high, resource language is used to help improve the results of the low-resource language (which has only 100/1000 sentences), and (3) HR column

[4] Accuracy is for the whole-tag ignoring the lemma. The F1 score is based on the precision and recall of each morphological feature ignoring the lemma, similar to Malaviya et al. (2018).

HR/LR	Model	LR100	XFER100	LR1000	XFER1000	HR
DA/SV	Cotterell	15.11	66.06	68.64	82.26	91.79
	Malaviya	29.47	63.22	71.32	77.43	
	Morse	62.45(0.69)	72.70(0.59)	86.44(0.17)	87.55(0.22)	92.68(0.19)
	MorseTag	**66.19(1.23)**	**76.70(0.72)**	**88.31(0.17)**	**88.97(0.54)**	**93.35(0.23)**
RU/BG	Cotterell	29.05	52.76	59.20	71.90	82.02
	Malaviya	27.81	46.89	39.25	67.56	
	Morse	59.82(1.65)	69.27(0.54)	87.71(0.26)	88.70(0.16)	85.43(0.12)
	MorseTag	**66.97(1.34)**	**75.78(0.26)**	**88.96(0.41)**	**90.52(0.21)**	**86.51(0.36)**
FI/HU	Cotterell	21.97	51.74	50.75	61.80	85.25
	Malaviya	33.32	45.41	45.90	63.93	
	Morse	49.58(1.27)	54.84(0.71)	**72.28(0.74)**	71.33(1.83)	**91.24(0.28)**
	MorseTag	**54.87(0.72)**	**57.12(0.36)**	**73.55(0.72)**	**73.86(1.28)**	**91.42(0.84)**
ES/PT	Cotterell	18.91	79.40	74.22	85.85	**93.09**
	Malaviya	58.82	77.75	76.26	85.02	
	Morse	**70.57(0.54)**	80.01(0.38)	**86.29(0.31)**	87.51(0.27)	**92.95(0.21)**
	MorseTag	**70.80(1.14)**	**81.60(0.16)**	**86.24(0.28)**	**88.01(0.13)**	**92.89(0.18)**

Table 5: Accuracy comparisons for UDv2.1 data sets. Table 6 gives F1 comparisons which are similar. LR is the low-resource language, HR is the high-resource language, XFER represents HR to LR transfer learning. 100/1000 indicate the number of sentences in the training set for low-resource experiments. Morse and MorseTag rows give the average of 5 experiments with standard deviation in parentheses. Statistically significant leaders ($p < 0.05$) are marked in *bold*. Some experiments have multiple leaders marked when their difference is not statistically significant.

HR/LR	Model	LR100	XFER100	LR1000	XFER1000	HR
DA/SV	Cotterell	08.36	73.95	76.36	87.88	94.18
	Malaviya	54.09	78.75	84.42	87.56	
	Morse	72.77(0.74)	81.39(0.27)	91.52(0.07)	92.42(0.15)	95.18(0.11)
	MorseTag	**74.91(1.26)**	**84.27(0.48)**	**92.39(0.26)**	**93.04(0.35)**	**95.50(0.21)**
RU/BG	Cotterell	14.32	58.41	67.22	77.89	90.63
	Malaviya	40.97	64.46	60.23	82.06	
	Morse	68.90(1.36)	76.86(0.41)	92.38(0.13)	93.12(0.21)	93.08(0.03)
	MorseTag	**75.52(1.16)**	**83.60(0.06)**	**93.08(0.37)**	**94.24(0.11)**	**93.55(0.13)**
FI/HU	Cotterell	13.30	68.15	58.68	75.96	90.54
	Malaviya	54.88	68.63	74.05	85.06	
	Morse	65.17(1.17)	71.77(0.42)	85.96(0.42)	85.91(0.86)	**95.34(0.20)**
	MorseTag	**72.21(0.67)**	**74.17(0.14)**	**87.17(0.38)**	**87.39(0.53)**	**95.37(0.52)**
ES/PT	Cotterell	07.10	86.03	81.62	91.91	**96.57**
	Malaviya	73.67	88.42	87.13	92.35	
	Morse	**80.06(0.73)**	88.11(0.25)	**92.43(0.28)**	93.31(0.20)	**96.52(0.10)**
	MorseTag	**80.07(0.92)**	**88.99(0.42)**	**92.29(0.28)**	**93.56(0.14)**	**96.44(0.13)**

Table 6: F1 comparisons for UDv2.1 data sets. See Table 5 for column descriptions.

gives the high-resource setting where we use the full training data with the high resource languages Danish, Russian, Finnish, and Spanish.[5]

For transfer experiments we use a simple transfer scheme: training with the high-resource language for 10 epochs and using the resulting model to initialize the compatible weights of the model for the low-resource language. All LSTM weights and embeddings for identical tokens are transferred exactly, new token embeddings are initialized randomly.

In all low-resource, transfer-learning, and high-resource experiments, Morse and MorseTag perform significantly better than the two related models (with the single exception of the high-resource experiment on Spanish, a morphologically simple

[5]Malaviya is missing from the HR column because we could not train it with large data sets in a reasonable amount of time. For Cotterell we used the SPECIFIC model given in Malaviya et al. (2018) in all experiments.

Method	TrMor2006	TrMor2016	TrMor2018
(Yuret and Türe, 2006)	95.82	-	-
(Sak et al., 2007)	96.28	-	-
(Yıldız et al., 2016)	-	92.20	-
(Shen et al., 2016)	96.41	-	-
Morse	95.94	92.63	97.67
MorseDisamb	96.52	**92.82**	**98.59**

Table 7: Test set lambda+tag accuracy of several models on Turkish data sets: TrMor2006 (Yuret and Türe, 2006), TrMor2016 (Yıldız et al., 2016), TrMor2018 (published with this paper).

language, where the difference with Cotterell is not statistically significant). This supports the hypothesis that the sequence decoder of Morse is more sample-efficient than a whole-tag model or a neural factor graph model.

Tag-only prediction in MorseTag generally outperforms joint lemma-tag prediction in Morse but the difference decreases or disappears with more training data and in simpler languages. In half of the high-resource experiments, their difference is not statistically significant. The difference is also insignificant in most of the experiments, with the morphologically simplest language pair Spanish/Portuguese.

5.3 Turkish Results

Table 7 shows the lemma+tag test accuracy of several systems for different Turkish data sets. We masked digits and `Prop` (proper noun) tags in our evaluations. The older models use a hand-built morphological analyzer (Oflazer, 1994) that gives a list of possible lemma+tag analyses and trains a disambiguator to pick the correct one in the given context. Standard Morse works without a list of analyses, the decoder can generate the lemma+tag from scratch. Older disambiguators always obtain 100% accuracy on unambiguous tokens with a single analysis, whereas Morse may fail to generate the correct lemma+tag pair. In order to make a fair comparison we also tested a version of Morse that disambiguates among a given set of analyses by comparing the probability assigned to them by the decoder (MorseDisamb).

MorseDisamb gives the best results across all three data sets. The best scores are printed in bold where the difference is statistically significant. None of the differences in TrMor2006 are statistically significant because of the small size of the test set. In TrMor2016 both Morse and MorseDisamb give state of the art results. The TrMor2018 results were obtained using an average of 5 random splits into 80%, 10%, and 10% for training, validation, and test sets.

Method	A	U	T
word	94.38	98.70	96.72
word+context	96.21	98.52	97.69
word+context+output	96.43	98.80	97.79

Table 8: Ablation analysis test set performances on the TrMor2018 data set. A: **A**mbiguous Accuracy, U: **U**nambiguous accuracy, T: **T**otal accuracy.

Note that the numbers for the three data sets are significantly different. Each result naturally reflects the remaining errors and biases in the corresponding data set, which might result in the true accuracy figure being higher or lower. Despite of these imperfections, we believe the new TrMor2018 data set will allow for better comparison of different models in terms of learning efficiency thanks to its larger size and lower noise level.

5.4 Ablation Analysis

In this section, the contributions of the individual components of the full model are analyzed. In the following three ablation studies, we disassemble or change individual modules to investigate the change in the performance of the model. We use the TrMor2018 data set in the first two experiments and UD data sets in the last experiment. Table 8 presents the results.

We start our ablation studies by removing both the context encoder and the output encoder, leaving only the word encoder. The resulting model (word) is a standard sequence-to-sequence model that only uses the characters in the target word without any context information. This gives us a baseline and shows that more than 95% of the wordforms can be correctly tagged ignoring the context.

	count=0			count<100			count≥100		
Lang	**Tok**	**Tag**	**Seq**	**Tok**	**Tag**	**Seq**	**Tok**	**Tag**	**Seq**
SV	12	0.0	8.33	844	81.28	82.82	19521	94.49	94.65
BG	4	0.0	0.0	910	81.32	83.41	14810	96.62	97.37
HU	108	0.0	20.37	2333	53.54	59.24	8007	78.24	80.67
PT	3	0.0	0.0	207	63.29	67.63	9991	93.04	92.25

Table 9: Test accuracy for tags that were observed 0, < 100, and ≥ 100 times in the 1000 sentence training sets. **Tok** is the number of tokens with the specified count, **Tag** is the accuracy using a whole-tag classifier, **Seq** is the accuracy using a sequence decoder.

	count=0		count<5		count≥5	
Dataset	**Tok**	**Acc**	**Tok**	**Acc**	**Tok**	**Acc**
TRMor2006	30	86.67	16	100.0	816	98.9
TRMor2016	79	2.53	579	93.78	18570	98.48
TRMor2018	0	-	1702	82.78	45119	99.48
UD-DA	1019	71.84	1023	94.72	7981	98.93
UD-ES	593	79.26	627	95.37	10780	99.36
UD-FI	2279	61.34	1802	88.85	16989	98.21
UD-RU	1656	77.48	1587	94.39	8305	99.22

Table 10: Test accuracy for lemmas that were observed 0, < 5, and ≥ 5 times in the TRMor and UD data sets. **Tok** is the number of tokens with the specified count, **Acc** is the accuracy using Morse.

We then improve the model by adding the context encoder (word+context). We observe a 1.83% increase in ambiguous word accuracy and 0.97% in overall accuracy. This version is capable of learning more than only a single morphologic analysis of each wordform. As an example, the lemma ''*röportaj*'' (interview) has 5 distinct wordforms observed in the training set. We tested both models on the never before seen wordform ''röportajı'' in ''*Benden bu röportajı yalanlamamı rica etti.*'' (I was asked to deny the interview). Whereas (word) failed by selecting the most frequently occurring tag of ''röportaj'' in the training set (`Noun+A3sg+Pnon+Nom`), word+context disambiguated the target wordform successfully (`+Noun+A3sg+Pnon+Acc`), demonstrating the ability to generalize to unseen wordforms.

Finally, we add the output encoder to reconstruct the full Morse model (word+context+output). We observe a further 0.22% increase in ambiguous word accuracy and 0.10% increase in overall accuracy. These experiments show that each of the model components have a positive contribution to the overall performance.

We believe our ablation models have several advantages over a standard sequence-to-sequence model: Both the input and the output of the system needs to be partly character based to analyze morphology and to output lemmas. This leads to long input and output sequences. By running the decoder separately for each word, we avoid the necessity to squeeze the information in the whole input sequence into a single vector. A standard sequence-to-sequence model would also be more difficult to evaluate as it may produce zero or multiple outputs for a single input token or produce outputs that are out of order. A per-word decoder avoids these alignment problems as well.

To compare our approach to whole-tag classifiers like Heigold et al. (2017), we created two versions of the (word+context) model, one with a sequence decoder and one with a whole-tag classifier. We trained these models on Turkish and UD data sets to test unseen/rare tag and lemma generation. Table 9 shows the accuracy of each model on three sets of tags: unseen tags, tags that were seen less than 100 times and tags that were seen at least 100 times in the training set. The sequence decoder generally performs better across different frequency ranges. In particular, results confirm that the sequence decoder can generate some unseen tags correctly while the whole-tag classifier in principle cannot. We observe that the advantage is smaller for more frequent tags, in fact the whole-tag classifier performs better with the most frequent tags in Portuguese, a morphologically simple language. A similar trend is observed in Table 10 for lemma generation:

Morse is able to generate a significant percent of the unseen/rare lemmas correctly.

6 Conclusion

In this paper, we presented Morse, a language-independent character-based encoder-decoder architecture for morphological analysis, and TrMor2018, a new Turkish morphology data set manually confirmed to have 96% inter-annotator agreement. The Morse encoder uses two different unidirectional LSTMs to obtain word and output embeddings and a bidirectional LSTM to obtain the context embedding of a target word. The Morse decoder outputs the lemma of the word one character at a time followed by the morphological tag, one feature at a time. We evaluated Morse on nine different languages, and obtained state-of-the-art results on all of them. We provided empirical evidence that producing morphological features as a sequence outperforms methods that produce whole tags or feature sets, and the advantage is more significant in low-resource settings.

To our knowledge, Morse is the first deep learning model that performs joint lemmatization and tagging, performs well with unknown and rare wordforms and tags, and can produce a variable number of features in multiple inflectional groups to represent derivations in morphologically complex languages.

Acknowledgments

We would like to thank Kemal Oflazer and all student annotators for their help in creating the TrMor2018 data set, and the editors and anonymous reviewers for their many helpful comments. This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) grants 114E628 and 215E201.

References

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. The CELEX lexical database (release 2). *Distributed by the Linguistic Data Consortium, University of Pennsylvania*.

Grzegorz Chrupala. 2006. Simple data-driven context-sensitive lemmatization. *Procesamiento del Lenguaje Natural*, 37.

Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *LREC 2008*.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen.

Daoud Daoud. 2009. Synchronized morphological and syntactic disambiguation for Arabic. *Advances in Computational Linguistics*, 41:73–86.

Turhan Daybelge and İlyas Çiçekli. 2007. A rule-based morphological disambiguator for Turkish. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), Borovets*, pages 145–149.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Gülşen Eryiğit and Kemal Oflazer. 2006. Statistical dependency parsing of Turkish. In *Proceedings of the 11th EACL*, pages 89–96. Trento.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Jan Hajič, Jan Votrubec, Pavel Krbec, Pavel Květoň. 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 67–74.

Dilek Zeynep Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410.

Dilek Zeynep Hakkani-Tür, Murat Saraçlar, Gökhan Tür, Kemal Oflazer, and Deniz Yuret. 2018, Morphological disambiguation for Turkish. In K. Oflazer and M. Saraçlar, editors, *Turkish Natural Language Processing*, Theory and Applications of Natural Language

Processing, chapter 3, Springer International Publishing.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 1, pages 505–513.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Arto Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, Walter de Gruyter & Co., Hawthorne, NJ.

Lauri Karttunen and Kent Wittenburg. 1983. A two-level morphological analysis of English. *Texas Linguistic Forum*, 22:217–228.

Kimmo Koskenniemi. 1981. An application of the two-level model to Finnish. *Computational Morphosyntax: Report on Research*, 1984:19–41.

Kimmo Koskenniemi. 1983. Two-level model for morphological analysis. In *IJCAI*, volume 83, pages 683–685.

Chaitanya Malaviya, Matthew R. Gormley, and Graham Neubig. 2018. Neural factor graph models for cross-lingual morphological tagging. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2653–2663.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, WA.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Kemal Oflazer. 2018, Morphological processing for Turkish. In K. Oflazer and M. Saraçlar, editors, *Turkish Natural Language Processing*, Theory and Applications of Natural Language Processing, Chapter 2, Springer International Publishing.

Kemal Oflazer, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 1999. Design for a Turkish treebank. In *Proceedings of the Workshop on Linguistically Interpreted Corpora, EACL 99*. Bergen.

Kemal Oflazer and İlker Kuruöz. 1994. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 144–149.

Kemal Oflazer and Gokhan Tür. 1996. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In *Conference on Empirical Methods in Natural Language Processing*.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007, Morphological disambiguation of Turkish text with perceptron algorithm. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007*, pages 107–118. Springer, Berlin Heidelberg.

Qinlan Shen, Daniel Clothiaux, Emily Tagtow, Patrick Littell, and Chris Dyer. 2016. The role of context in neural morphological disambiguation. In *COLING*, pages 181–191.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA.

Alymzhan Toleu, Gulmira Tolegen, and Aibek Makazhanov. 2017. Character-aware neural morphological disambiguation. In *Proceedings*

of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 666–671.

Eray Yıldız, Çağlar Tirkaz, H. Bahadır Şahin, Mustafa Tolga Eren, and Ozan Sönmez. 2016. A morphology-aware network for morphological disambiguation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2863–2869.

Deniz Yuret. 2016. Knet: Beginning deep learning with 100 lines of julia. In *Machine Learning Systems Workshop at NIPS 2016*.

Deniz Yuret and Ferhan Türe. 2006. Learning morphological disambiguation rules for Turkish. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 328–334.