Morphological Tagging

Yash Kumar Lal, SBU CS

- Don’t Throw Away Those Morphological Analyzers Just Yet: Neural Morphological Disambiguation for Arabic. Zalmout and Habash, EMNLP 2017

- Joint Diacritization, Lemmatization, Normalization, and Fine-Grained Morphological Tagging. Zalmout and Habash, ACL 2020

Don’t Throw Those Morphological Analyzers Away Just Yet: Neural Morphological Disambiguation for Arabic

Nasser Zalmout and Nizar Habash, EMNLP 2017

Overview

- BiLSTM models for morphological tagging and language modeling
- Models used to rank outputs of a morphological analyzer
- Word and character based embeddings
- Incorporate subword and morphological features at different depths in tagger
- 4.4% accuracy gain over SotA; 10.6% for OoV words

Issues due to MRLs

- Higher ambiguity due to different interpretations of same surface form
- MSA - optional diacritization in orthography (avg 12 analyses per word)
- Richness of form -> higher model sparsity
- Limited and noisy data

	MSA	En
Tokens	0.8x	x
Types	2y	y

Feature	Definition
diac	Diacratization
lex	Lemma
pos	Basic part-of-speech tags (34 tags)
gen	Gender
num	Number
cas	Case
stt	State
per	Person
asp	Aspect
mod	Mood
vox	Voice
prc0	Proclitic 0, article proclitic
prc1	Proclitic 1, preposition proclitic
prc2	Proclitic 2, conjunction proclitic
prc3	Proclitic 3, question proclitic
enc0	Enclitic

Table 1: The morphological features we use in the various models. The first two groups are lexical features; and the last two groups are inflectional and clitic features respectively, in addition to the part-of-speech tag.

Dataset

- Penn Arabic Treebank; follow Diab et al, 2013's splits
- Alif/Ya and Hamza normalization
- Remove diacritization
- Pretrained word embeddings
  - Word2Vec (Mikolov et al, 2013a,b)
  - Learned over LDC's Gigaword MSA corpus

	Words (size)
Train	503,015
Dev	63,137
Test	63,172
Gigaword	2,154M

Baselines

- Maximum Likelihood Estimator - count frequency scores for each word/tag out of context, with backoff to most frequent word/tag for unknown words
- MADAMIRA w/ ADD_PROP - MADAMIRA w/ proper noun analysis for all words

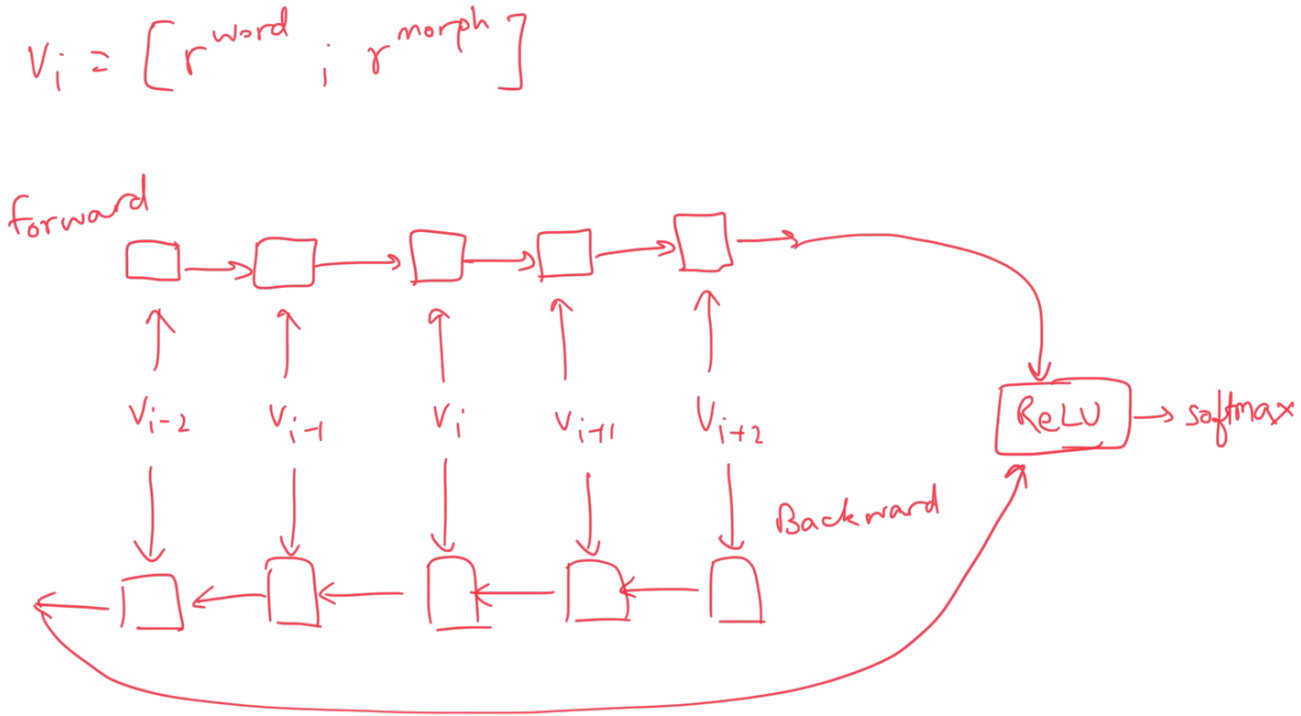
Evaluation

- Accuracy-based metrics
- EVALFULL - correctly analyzed all morphological features
- EVALDIAC - correct fully diacritized form
- EVALLEX - correct lemma
- EVALPOS - correct PoS tag
- EVALATBTOK - correct ATB tokenization
- Morphological disambiguation models tend to struggle with the metrics in the order listed (higher accuracy as you move down the list)

Modeling

- Task - Choose correct morphological analysis from analyzer’s output set
- Categorize types of features into morphological and lexical
- Morphological features - Tagger to obtain relevant tag
- Lexical features - Language modeling
- 14 different taggers, one per feature
- BiLSTMs for each lexical feature
- Combine all for morphological analysis

Morphological Tagger



Subword & Morphological features

- Fixed width affixes - 3 characters at start and end of word
- Language specific affixes - Regex to maximally match affix patterns at start and end of word; use lightstemmer
- PoS tags from morphological dictionaries - concatenate potential tags with word embedding

$$r_c^{morph} = r_1^{morph} + r_2^{morph} + \dots$$

← Sum for each individual component

r_2^{morph} → one-hot vector

- Character embedding is concatenation of word embedding with one-hot embedding of each character in word

PoS tagging - word vs char embeddings

Model	**Embedding**	
	Word	Char
No Morphology	96.4	96.7
Fixed Character Affixes	96.6	NA
Lightstemmer	96.7	96.8
Morphological Dictionary	97.5	97.5
+ Fixed Character Affixes	**97.6**	NA
+ Lightstemmer	**97.6**	**97.6**

PoS tagging - error analysis

- Categorize tagset - nominals, verbs, particles, punctuations
- Most errors - nominals
- Effect of morphological dictionary - verbs have highest accuracy increase
    - May be because they are least frequent category

Morphological Disambiguation

- Apply a tagger to all morphological features
- Language models for lexical features
- Result of these acts as input to scorer and ranker

Task 1 - Morphological Tagging

- Model - word embeddings w/ morphological dictionary, fixed/lightstemmer affixes
- One tagger for each feature

System	pos	cas	num	gen	vox	mod	stt	asp	per	enc0	prc0	prc1	prc2	prc3
MLE	92.5	80.5	98.3	97.5	97.7	97.4	90.2	97.9	97.9	98.3	97.9	98.5	97.9	99.6
MADAMIRA	97.0	91.1	99.5	99.4	99.1	99.1	97.0	99.3	99.2	99.6	99.6	99.6	99.6	**99.9**
Bi-LSTM	97.6	94.5	99.6	99.5	99.2	99.4	97.9	99.4	99.4	99.7	99.7	99.8	**99.7**	**99.9**
Disambiguated Bi-LSTM	**97.9**	**94.8**	**99.7**	**99.7**	**99.4**	**99.6**	**98.3**	**99.6**	**99.6**	**99.8**	**99.8**	**99.9**	**99.7**	**99.9**
Absolute Increase	0.9	3.7	0.2	0.3	0.3	0.5	1.3	0.3	0.4	0.2	0.2	0.3	0.1	0.0
Error Reduction	30.0	42.0	40.0	50.0	33.0	56.0	43.0	43.0	50.0	50.0	50.0	75.0	25.0	0.0

Table 6: Morphological tagging results. The absolute increase and error reduction are of the disambiguated Bi-LSTM against MADAMIRA.

Task 2 - Language Modeling

- For lemmatization and diacritization
- LSTM based model with class input
- Types, not tokens, to reduce vocabulary size
- Given the class of a word, predict the class of the next word
- Test set in HTK Standard Lattice Format - lattice contains different representations of each word

Feature	*lex*	*diac*
3-gram model	76.7	68.2
3-gram model disambiguated	96.2	87.7
Our system (LSTM)	89.6	73.5
Our system disambiguated	**96.9**	**91.7**

Table 7: The language model accuracy scores for both MADAMIRA and the LSTM models, for the *lex* and *diac* features.

Final Task - Morphological Disambiguation

- Simple regression style scorer
- Tagger - class output; LMs - class output
- If the analysis and the predicted morph tag for a word match, weight for that analysis is increased
- Result = analysis with highest score
- Feature weight tuning using Simplex method

- Retrain all taggers and LMs on full training set using optimal weight
- Remember, EVALFULL = Diac + Lex + PoS + Tok

Evalustion Metric	All Words			Out-Of-Vocabulary Words		
	MADAMIRA	Our System	Error Reduction	MADAMIRA	Our System	Error Reduction
EVALFULL	85.6	**90.0**	30.6	66.3	**76.9**	31.5
EVALDIAC	87.7	**91.7**	32.5	70.2	**79.8**	32.8
EVALLEX	96.2	**96.8**	15.8	82.9	**87.8**	28.7
EVALPOS	97.0	**97.9**	30.0	89.9	**96.0**	60.4
EVALATBTOK	99.4	**99.6**	33.3	94.2	**97.8**	60.2

Table 8: Accuracy results of the disambiguation system, evaluated using different metrics, for all words and out-of-vacbulary (OOV) words alone. OOV percentage of all words is 7.9%.

Error Analysis

	LSTM	MADAMIRA
Case	N	Y
Mood	Y	N
Voice	N	N
Syntax	Y	N
2nd person cases	N	-

Y indicates that model does better than the other; N indicates the opposite; Both Ns indicate that both models don’t work well

Future Work

- Other deep learning architectures, especially joint modeling approaches and seq2seq
- Exploit character level embeddings
- Role of syntax features

Joint Diacritization, Lemmatization, Normalization, and Fine-Grained Morphological Tagging

Nasser Zalmout and Nizar Habash, ACL 2020

Overview

- Joint modeling of lexicalized (character level) and non-lexicalized (word level) features
- Seq2seq architecture w/ different parameter sharing strategies in encoder & decoder for different feature types
- Non-lexicalized features modeled by tagger in multi-task framework
  - Some parameters shared with encoder
- Lexicalized features have same encoder and different decoder
  - Fixed context window on character level

Modeling

1. Tagger
2. Encoder
3. Decoder

Tagger

- Word vectors using FastText on external data (w)
- LSTM over characters of word; last state is representation (s)
- For each feature, obtain all possible values using morph analyzer
  - Separate embedding tensors learnt for each feature
- Sum all vectors to get morph tag vector (a_f) for one feature
- Concatenate a_f for all features to get final morph tag vector (a)

$v_j = [w_j ; s_j ; a_j]$

w – word ; s – character
a - morphological tags

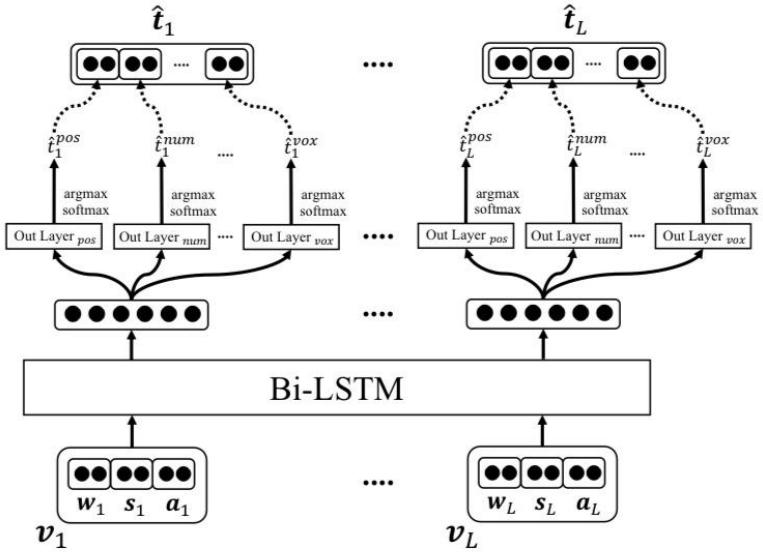


Figure 1: The tagger model, showing the multitask learning architecture for the features. The concatenated predicted tags are used to condition on, at the decoders.

Encoder

- Shares word & character embeddings with tagger
- Input context - sliding window of fixed number of characters around target word

$$c_i \rightarrow [c_i ; w_j]$$

$$\text{Objective} = P(y_k | c_i, w_j)$$

Decoder

- Separate decoder for lemmatization and diacritization
- Condition on previous decoder output, last encoder output and tags predicted by tagger
- Decoder output -> softmax -> argmax -> character at each time step

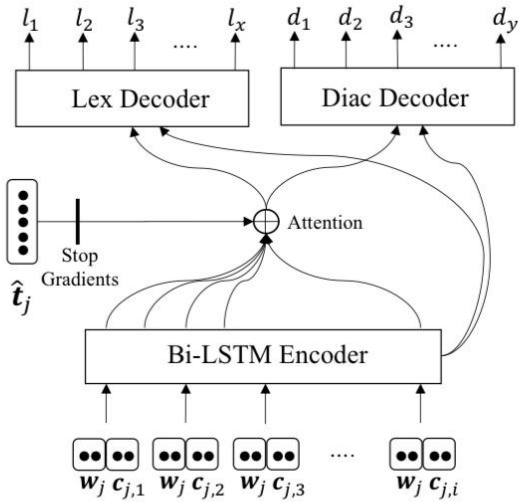


Figure 2: The sequence-to-sequence architecture for the lexicalized features, with a shared encoder, and separate decoders for lemmatization and diacritization. The figure does not show the fixed context window of 10 characters before and after the target word.

$$H(\hat{y}, y) = \frac{1}{|F|} \sum_{f \in F} H(\hat{y}^f, y^f)$$

Training Details

- No validation set - tuned on 5% of train set
- Train for fixed number of epochs, select best model
- Adam optimizer
- LR = 0.0005
- 50 epochs of training

Dataset

- PATB for MSA, ARZ dataset for EGY
- EGY and MSA processed similarly (as described in Zalmout and Habash, EMNLP 2017)
- Morphological Analyzers
  - MSA - SAMA
  - EGY - SAMA + CALIMA + ADAM
- Word Embeddings data
  - MSA - LDC Gigaword
  - EGY - BOLT Arabic Forum Discussions corpus

Evaluation

- POS - Accuracy of PoS tag prediction (out of 36 tags)
- TAGS - Accuracy of all (14) morphological features
- DIAC - Correctly predict only diacritized forms (MSA only)
- CODA - CODA normalization for EGY
- LEMMA - Lemma prediction accuracy
- FULL - Accuracy of full morphological analysis

Baselines

1. MADAMIRA
2. Zalmout and Habash, EMNLP 2017
3. Zalmout and Habash, ACL 2019

Results

	Model	FULL	TAGS	DIAC	LEX	POS
MSA	(a) MADAMIRA (SVM models + analyzer) (Pasha et al., 2014)	85.6	87.1	87.7	96.3	97.1
	(b) LSTM models + analyzer (Zalmout and Habash, 2017)	90.4	92.3	92.4	96.9	97.9
	(c) + Multitask learning for the tags (Zalmout and Habash, 2019)	90.8	92.7	92.7	96.9	97.9
	(d) Joint modeling + analyzer	**92.3**	**93.5**	**93.9**	**97.6**	**98.1**
	(e) Joint modeling without analyzer	90.3	92.7	92.8	96.3	97.7

	Model	FULL	TAGS	CODA	LEX	POS
EGY	(a) MADAMIRA (SVM models + analyzer) (Pasha et al., 2014)	76.2	86.7	82.4	86.4	91.7
	(b) LSTM models + analyzer (Zalmout and Habash, 2017)	77.0	88.8	82.9	87.6	92.9
	(c) + Multitask learning for the tags (Zalmout and Habash, 2019)	77.2	88.8	82.9	87.6	**93.1**
	(d) Joint modeling + analyzer	**79.5**	**89.0**	**85.0**	**88.5**	**93.1**
	(e) Joint modeling without analyzer	73.2	84.9	81.5	84.4	91.1

Table 3: The results of the various models on the **DEVTEST** for MSA and EGY. The first and second baselines, (a) and (b), use separate models for the features, and the third, (c), uses a multitask learning architecture for the non-lexicalized features only.

Results (contd.)

- Diacritization
  - Helped most by joint modeling
  - Baseline has large target space for diacritized forms as compared to other tasks (like lemmatization) so it’s performance is poor
- Normalization
  - Usually a byproduct of the approach, not the goal
  - Here, character level models have explicit normalization capability
- Consistent Analysis
  - If all features are linguistically acceptable to co-occur with each other
  - E.g., Verbs don’t have case. So, if verb analysis has a case value, it is inconsistent

Effect of Morphological Analyzer

- Adds additional consistency b/w different features no matter how good a model already is - Explicit knowledge helps in all cases
- 110 errors in sample of 1000 MSA DEVTEST examples
- 62% - consistent feature prediction which didn’t match gold prediction
- 13% - gold errors
- 25% - inconsistent predictions
- So, accuracy boost due to morphological analyzer can be attributed to its consistency

Joint vs Separate Modeling

- Investigate distribution of errors for both approaches
- Annotate 1000 word sample w/ main erroneous feature
- Joint model does better for diacritization
- For EGY, prediction sometimes matches MSA standard word
  - Likely due to code switching in dialectal content)
- Gold errors are frequent

Future Work

- Consistency of predicted features without morphological analyzers

Questions?