Arabic diacritization: Classical ML, RNNs, and Seq2seq models

KAREEM DARWISH

Farasa Arabic NLP Toolkit

Farasa.qcri.org

Farasa: Arabic NLP

Comprehensive toolkit for Arabic processing:

- Word segmentation: wktAbhm ➔ w+ktAb+hm (وكتابهم : و+كتاب+هم)
- Lemmatization: ktAb (كتاب)
- Part-of-speech tagging: CC+NOUN+PRON
- Named entity recognition: PERS, LOC, ORG
- Parsing
- Diacritic recovery
- Spell checking

Farasa is state-of-the-art in all of its components while being blistering fast.

Arabic Diacritization

KAREEM DARWISH
WITH: HAMDY MUBARAK, AHMED ABDELALI

Context and Motivation

Modern Standard Arabic (MSA) text is typically written without diacritics (short vowels) Diacritics are essential for proper disambiguation/pronunciation of words

Ex. علم (Elm) → عَلَم (Ealam - flag), عِلْم (Eilom - knowledge), etc.

Diacritic recovery is critical for text-to-speech and pedagogy

Farasa, our diacritizer:

disambiguates ***word-cores*** (based on context) – ex. عَلَم

determines ***case-endings*** (based on syntactic role) – ex. علمٌ

Core word diacritics

ARABIC DIACRITIZATION

Core Word Diacritics

Core word diacritic recovery requires disambiguation of words in context:

Ex.

*hbt <lY *Almdrsp* (ذهبت إلى **المدرسة**) – I went (she went) to *the school*

qAblt *Almdrsp* (قابلت **المدرسة**) – I met *the (female) teacher*

Core Word Diacritics: Training Data

We acquired a diacritized corpus from a commercial vendor:

- contains 9.7 million tokens (194k unique undiacritized tokens)
- composed mostly of MSA (≈ 7M tokens) and some religious text (≈ 2.7M tokens)
- covers multiple genres: politics, economics, sports, science, etc.
- has an estimated diacritization errors < 1%
- no omissions of sukun or optional diacritics.

Ask me why we didn't use ATB

Core Word Diacritics: Dictionary

Given every word, we produce multiple representations (ex. وكتابهم – wakitaAbihimo):

- full diacritized surface form (وَكِتَابِهِمْ – wakitaAbihimo)
- full diacritized surface form w/o case ending (وَكِتَابهمْ – wakitaAbhimo)
- diacritized stem w/ and w/o case ending (كِتَابِ – kitaAbi and كِتَاب – kitaAb)
- diacritized template of full form w/ and w/o case ending (وَفِعَالِهِمْ – wafiEaAlihimo and وَفِعَالهمْ – wafiEaAlhimo)
- diacritized stem template w/ and w/o case ending (فِعَالِ – fiEaAli and فِعَال – fiEaAl)

We built dictionaries from the different representations and unigram/bigram language models for words and stems

الكلمة	وكتباهم
مشكلة	وَكِتَابِهِمْ
مشكلة بدون حركة الإعراب	وَكِتَابهمْ
الجذع مشكل	كِتَابِ
الجذع مشكل دون حركة الإعراب	كِتَاب
التصريف مشكل	وَفِعَالِهِمْ
التصريف مشكل دون حركة الإعراب	وَفِعَالهم
تصريف الجذع مشكل	فِعَالِ
تصريف الجذع مشكل دون حركة الإعراب	فِعَال

Core Word Diacritics: Disambiguation

Baseline model uses a simple Hidden Markov Model (HMM) with a bigram Language Model:

- LM trained on surface form w/o case ending
- For a given sentence, we build a lattice of possible diacritized forms
- Example: مسح النص (msH AlnS – “deletion of the text” or “he deleted the text”, “the text was deleted”)
  - مسح (msH) → {مَسْح masoH, مَسَح masaH, مُسِح musiH}
  - النص (AlnS) → {النَّص Aaln~aS}
  - Viterbi algorithm picks مَسْح النَّص (masoH Aaln~aS – “deletion of the text”)
- Testing was performed on WikiNews test set – 70 articles, 7 genres, 18,300 words, and recent (2013 & 2014)

Core Word Diacritics: Defaults

- We constructed a dictionary of function words (ex. كَيْفَ (kyf – how)) and their diacritized forms
  - Some function words may be confused with other words (ex. عَنْ (Eano – from) and عَنَّ (Ean~a – appeared))
  - We assume they are function word, because they are far more common
- We constructed a dictionary of all words appearing more than 10 times, where one diacritized forms appears more than 90% of the time
  - Ex. غَزَّة (gaz~apa – Gaza)
  - غَزَّة usually appears as part of the collocation قِطَاع غَزَّة (qTAE gzp – Gaza Strip)

Core Word Diacritics: Back-offs

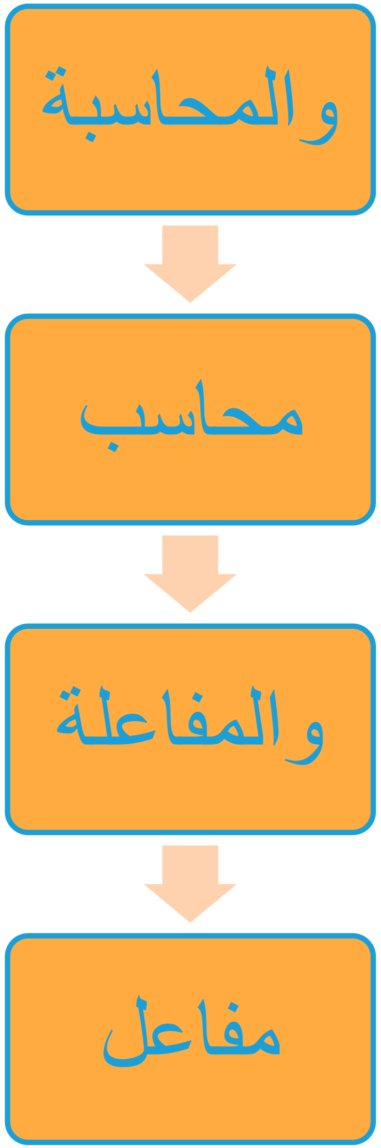
If a word is an OOV, we back-off to:

Stem: we stem the word and use the most likely diacritized form (Farasa Segmenter).

Ex. والمحاسبة (wAlmHAsbp – and the accountant (feminine)) → (و+ال w+Al) محاسب mHAsb (+ة +p) → مُحَاسِب (muHAsib)

Template: we get the template of the word and use the most likely diacritized form. If that fails, we back-off to stem template, and use the most likely one.

Ex. والمحاسبة (wAlmHAsbp) → (wAlmfAElp –→ waAalomufaAEilap & mfAEl –→ mufaAEil)



Core Word Diacritics: Transliteration

We used Transliteration Mining (TM) to learn diacritized forms of Arabic words (typically named entities) from English transliterations

Given a diacritized Arabic ↔ English transliteration pair, we obtain TM alignments:

Ex. حسن (Hsn) ↔ Hassan → {حَ (Ha) ↔ Ha, سَ (sa) ↔ ssa, نْ (no) ↔ n}

Train a Conditional Random Fields (CRF) sequence labeling model to learn diacritics from the Arabic-English-diacritic tuple {ح-Ha-a, س-ssa-a, ن-n-o}

Mohamed	محمد
Mo	مُ
Ha	حَ
Me	مَّ
D	د

Muhammad	محمد
Mu	مُ
Ha	حَ
Mma	مَّ
D	د

Core Word Diacritics: Transliteration

We trained using 3,452 diacritized Arabic ↔ English diacritized pairs
We extracted transliteration pairs from cross-lingual Wikipedia titles → 125k pairs
We applied the CRF model, leading to the diacritization of 68k Arabic words
Diacritization accuracy is 79%.

Core Word Diacritics: Results

System	% WER	% DER
Baseline	6.64	2.40
Defaults	4.54	1.75
Stem Back-off	4.69	1.44
Template Back-off	5.96	1.90
Transliteration Back-off	6.56	2.39
All Back-offs	4.51	1.35
Defaults+Back-offs	**3.29**	**1.06**
MADAMIRA	6.73	1.91
Rashwan et al. (2015)	***3.04***	***0.95***
Belinkov and Glass (2015)	14.87	3.89

Case Endings

ARABIC DIACRITIZATION

Case Endings: Parsing

Parsing (إعراب) is slow and SOTA Arabic parser (Farasa) has an accuracy of 89%.

Case Endings: Classical ML (SVM^{Rank})

We framed case ending recovery as a ranking problem

- Given possible case endings for a word, rank them using SVM^{Rank}

We used many features such as:

- current word and stem
- current POS tag, gender, and number (Farasa POS tagger)
- previous and next words, stems, and POS tags
- current word prefix(es) and suffix(es)
- current word and stem template
- complex features such as word bigrams and POS trigrams

Accounting for all features is HARD

Word	وكتابهم
Stem	كتاب
POS	cc+noun+pron
Gender	Masculine
Number	Single
Prefix	و+
Suffix	+هم
Template	وفعالهم، فعال

Case Endings: Heuristics

We used some heuristics to restrict the case endings that is SVM^{Rank} would rank, such as:

- If a word or POS appear more than 1,000 and 50 times respectively, restrict case endings to those seen in training
- If POS is a VERB, restrict to {a, o, u, ~a, ~u, or null} and to {a, o, ~a, or null} if not present tense.
- Restrict case endings for some suffixes (ex. “wn” → {a}.
- If stem POS is NOUN and more than 80% of time the case ending was “o” in the training or was diacritized using TM, then restrict to {o} – typically a named entity.
- Plus many more …

Case Ending: Classical ML Results

System	% WER	% DER
SVM^{Rank}	13.38	3.98
$\boldsymbol{SVM^{Rank} + Heuristics}$	***12.76***	***3.54***
MADAMIRA	19.02	5.42
Rashwan et al. (2015)	15.95	4.29
Belinkov and Glass (2015)	30.50	7.89

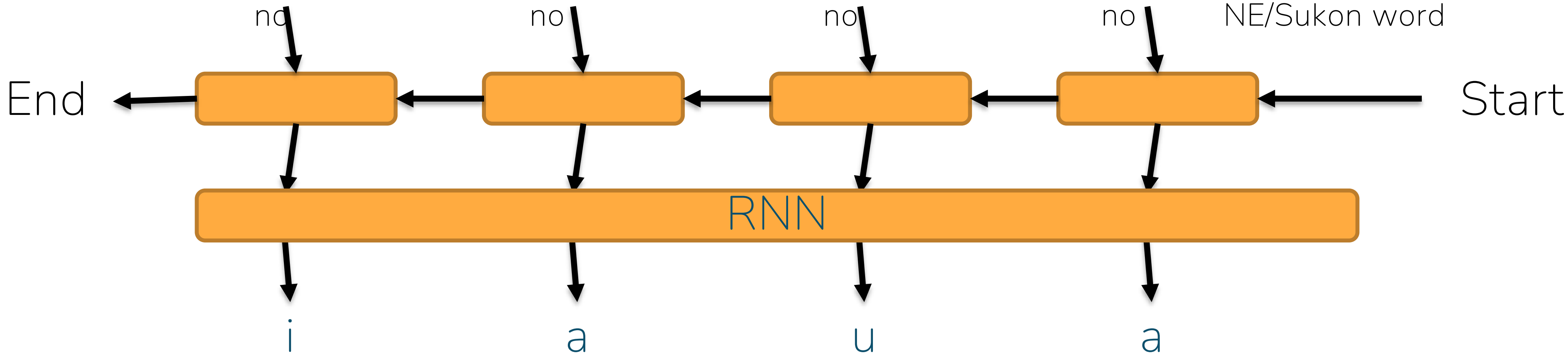
Case Endings: Recurrent Neural Networks

RNN’s account for ALL combinations of features
RNN’s can look at context

Word	و+رصد+ت	ال+باحث+ة	خلال	ال+دراس+ة
Stem	rSd	bAHv+p	KlAl	drAs+p
POS	CC+V+PRON	ال_التعريف+اسم+لاحق_اسم	NOUN	ال_التعريف+اسم+لاحق_اسم
Stem POS	V	NOUN+NOUN_Suff	NOUN	NOUN+NOUN_Suff
Stem template	fEl	fAEl	fEAl	fEAl+p
Prefixes	W+	Al+	-	Al+
Prefixes POS	CC	DET	-	DET
Suffixes	+t	-	-	-
Suffixes POS	PRON	-	-	-
Gender & Number	-	S, F	S, M	S, F
Leading characters	w, wr, wrS	A, Al, Alb	K, Kl, KlA	A, Al, Ald
Trailing characters	t, dt, Sdt	p, vp, Hvp	L, Al, lAl	p, sp, Asp
NE/Sukon word	no	no	no	no

Case Endings: RNNs

ال+دراس+ة	خلال	ال+باحث+ة	و+رصد+ت	Word
drAs+p	KlAl	bAHv+p	rSd	Stem
ال_التعريف+اسم+لاحق_اسم	NOUN	ال_التعريف+اسم+لاحق_اسم	CC+V+PRON	POS
NOUN+NOUN_Suff	NOUN	NOUN+NOUN_Suff	V	Stem POS
fEAl+p	fEAl	fAEl	fEl	Stem template
Al+	-	Al+	W+	Prefixes
DET	-	DET	CC	Prefixes POS
-	-	-	+t	Suffixes
-	-	-	PRON	Suffixes POS
S, F	S, M	S, F	-	Gender & Number
A, Al, Ald	K, Kl, KlA	A, Al, Alb	w, wr, wrS	Leading characters
p, sp, Asp	L, Al, lAl	p, vp, Hvp	t, dt, Sdt	Trailing characters
no	no	no	no	NE/Sukon word

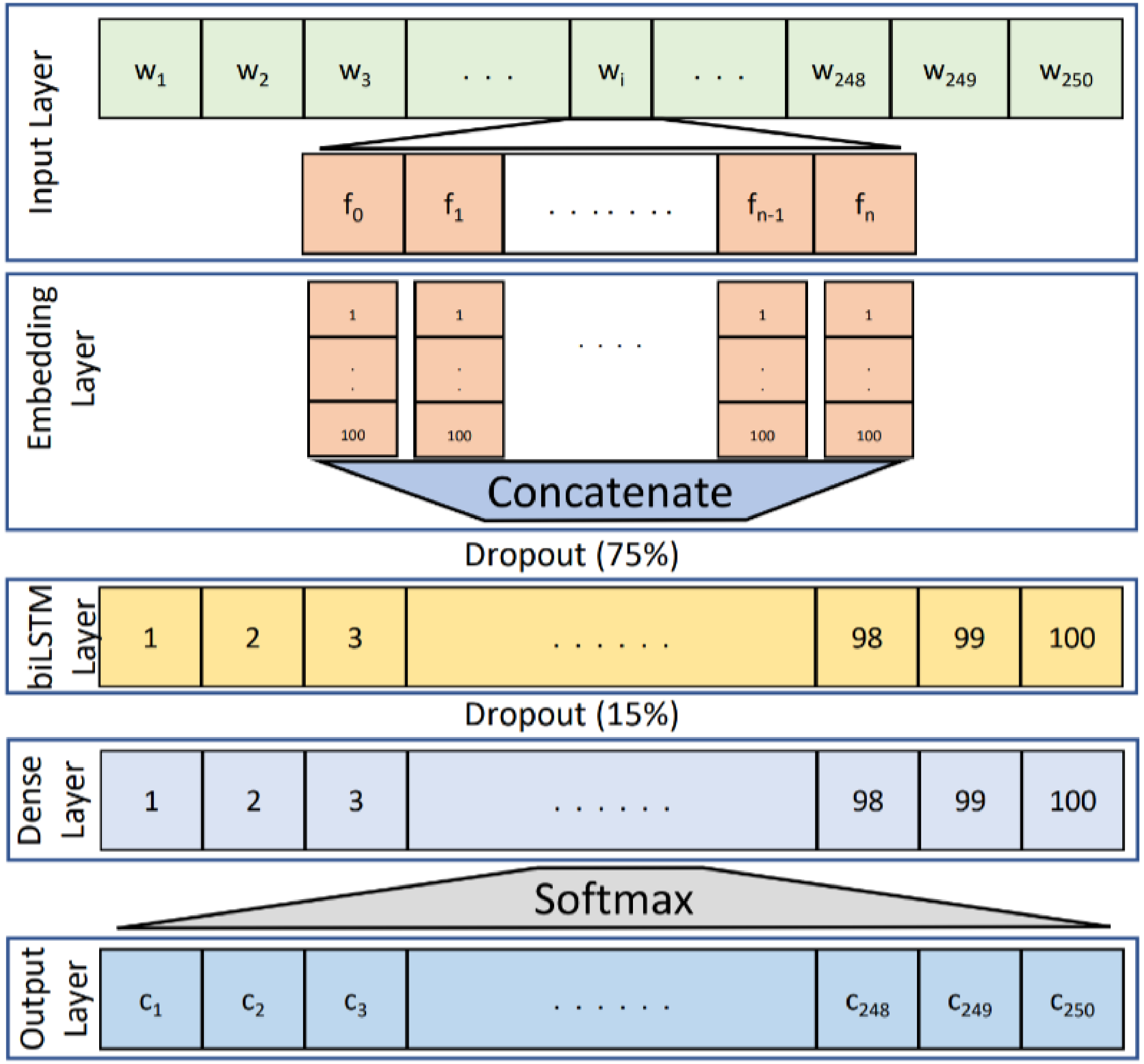


Case Endings: RNNs

RNNs can examine all possible combinations of features

RNNs are not tied to "human" logic, so it can examine features we never thought about

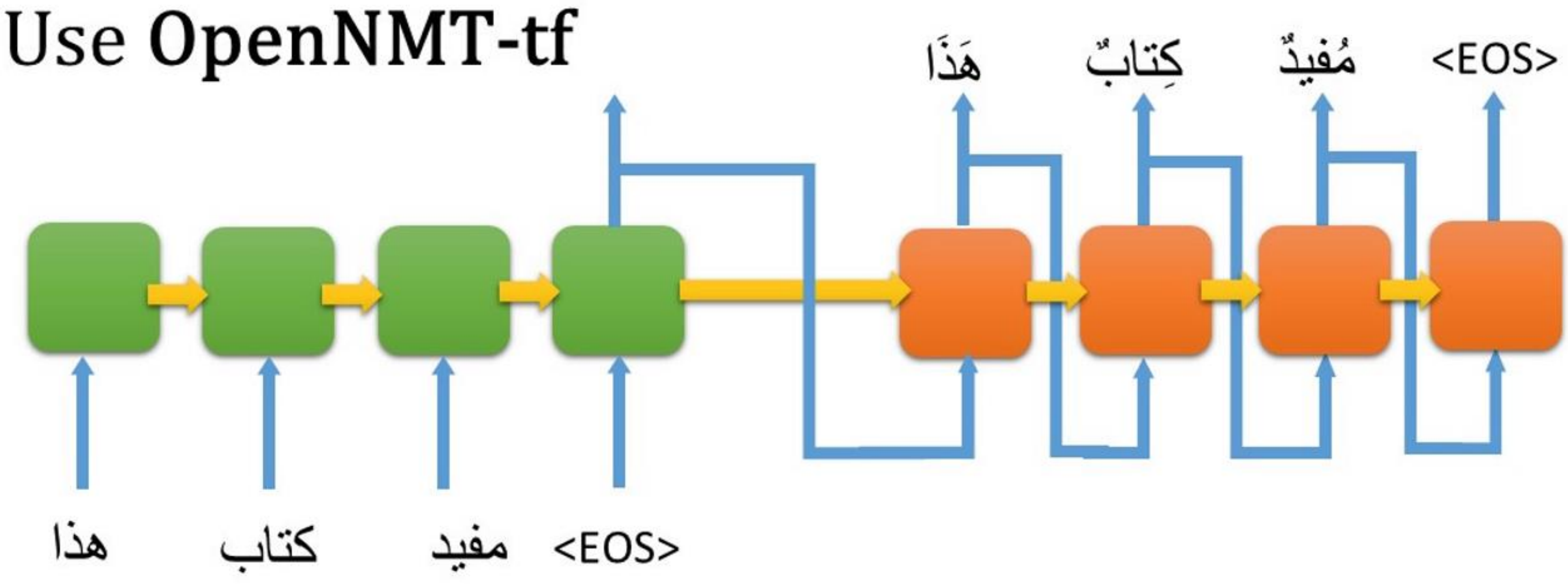
RNNs determine all case endings at the same time



Case Ending: RNN Results

Setup	CEER%
MSA	
word (baseline)	9.1
word-surface	5.7
word-POS	7.0
word-morph	7.6
word-surface-POS-morph	5.2
all-misc	**3.7**
Microsoft ATKS	9.5
Farasa	10.4
RDI [39]	14.0
MIT [9]	15.3
MADAMIRA [38]	15.9

Case Endings: Character Seq2Seq



Case Endings: Character Seq2Seq

Example: ...تمكن علماء بريطانيون من الوصول إلى بعد جديد في معرفة

ت م ك ن _ ع ل م ا ء _ ب ر ي ط ا ن ي و ن _ م ن ...

تَ مَ كَّ نَ _ عُ لَ مَ ا ءُ _ بِ رِ ي طَ ا نِّ يُ و نَ _ مِ نْ ...

Words are represented as characters
No need for feature engineering

WER = 48.3% ☹

Case Endings: Character Seq2Seq

1st trick: Limit context to a few words

2nd trick: Use multiple contexts, and we vote

3rd trick: Combine multiple context lengths

Case Ending: Seq2Seq Results

Description	Core WER%	CE WER%	WER%
Baseline Word	44.29	54.95	54.31
Baseline Char	41.29	41.95	48.31
Word 7g	14.83	19.01	20.69
Char 7g	2.78	6.11	8.32
Word 7g+overlap	14.50	16.57	18.05
Char 7g+overlap	2.04	3.23	4.94
Char 3g+overlap+voting	2.31	5.97	7.79
Char 5g+overlap+voting	2.37	3.57	5.49
Char 7g+overlap+voting	1.99	3.07	4.77
Char 11g+overlap+voting	3.03	3.93	6.40
Char 7g+overlap+voting (Transformer)	2.05	3.04	4.77
Combination $^{*}09 +^{\dagger} 11$	**1.89**	**2.89**	**4.49**

Classical ML vs. RNNs vs. Seq2Seq

We employ RNNs in production, because:

RNNs are much faster than seq2seq

Seq2seq tend to hallucinate

Accuracy of seq2seq > RNNs >>>> classical ML

Thank you – Questions?