KFTT

Polish Full Neural Morphosyntactic Tagger

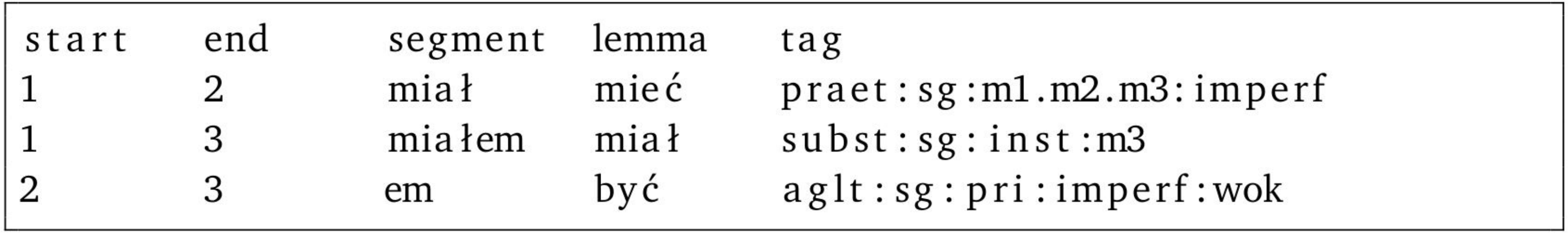
AGH

Krzysztof Wróbel

Good news

1. achieves 97.3% accuracy for contemporary texts
2. solves the problem with word “miałem”

Listing 1.2: Output from Morfeusz for word *miałem*.



```
start    end    segment   lemma   tag
1        2      miał      mieć    praet:sg:m1.m2.m3:imperf
1        3      miałem    miał    subst:sg:inst:m3
2        3      em        być     aglt:sg:pri:imperf:wok
```

Task

PolEval 2020 Task 2: Morphosyntactic tagging of Middle, New and Modern Polish

Data

- annotated using a historical tagset similar to Morfeusz SGJP
- represented as directed acyclic graphs of interpretations
- annotated by the date of creation
- not split into sentences

Table 2: Distribution of texts by time in training, development, and test data.

Subcorpus	Period	train	devel	test
KorBa — a corpus of 17th and 18th century	Middle	28.3%	50.0%	50.0%
a corpus of 19th century	New	42.6%	30.0%	30.0%
1M subcorpus of the National Corpus of Polish NKJP	Modern	29.1%	20.0%	20.0%

Table 1: Number of texts, tokens, the average number of tokens in texts, and the number of unique tags for training, development, and test data.

	train	devel	test
number of texts	10 755	244	280
number of tokens	1 441 508	40 016	40 045
average number of tokens in text	134	164	143
unique tags	994	571	582

Methods

Two separate steps:

- tokenization - most work
- tagging

Tokenization

The network answers a question if after every character should be the end of the token.

- forward and backward character-based language model using recurrent neural networks (RNN)
- bidirectional RNN
- conditional random field (CRF)

First version (`wo_morf`) uses only characters.

Second version uses exploits information from Morfeusz by appending to each character additional information, i.e. potential end of token, potential tags, and time of creation.

Listing 1.1: Output from Morfeusz with Baroque dictionary for word *zaś*.

```
start   end     segment  lemma    tag
1       2       za       za       part
1       3       zaś      zaś      conj
1       3       zaś      zaś      part
2       3       ś        być      aglt:sg:sec:imperf:nwok  nps
```

Table 3: Additional features generated for characters in word *zaś*.

Features	z	a	ś
is space before	True	False	False
joined tags	-	part	aglt:sg:sec:imperf:nwok_conj_part
joined POS	-	part	aglt_conj_part
century	17	17	17
is ambiguous	False	True	False

Tagging

- operates on tokenized text
- transformer model with a standard token classification head

Evaluation

Tokenization is measured on token level using precision, recall and F1.

The main metric in the competition is an accuracy -- a percentage of all tokens that match tagger segmentation with the correct tag.

The accuracy is also provided for known and unknown tokens for a morphological analyzer.

Additionally, the organizers report Acc on manual -- accuracy for manually tokenized words and manually appended correct interpretations to interpretations from the analyzer.

Experiments

The training was performed using only data provided by organizers.

The tokenization module uses Flair embeddings. The training lasts 24 hours on GPU Tesla V100 with a learning rate 0.1 and a hidden size of RNN 256.

For the tagging module, the transformer model has been chosen as a multi-language XLM-RoBERTa large version. The model was fine-tuned for 20 epochs using learning rate 5e-5, maximum sequence length 512, max gradient norm 1.0, without warmup steps. The training takes 4 hours using GPU Tesla V100.

Two versions were trained: using only training data (`train`) and using training and development data (`train+devel`).

Results - tokenization

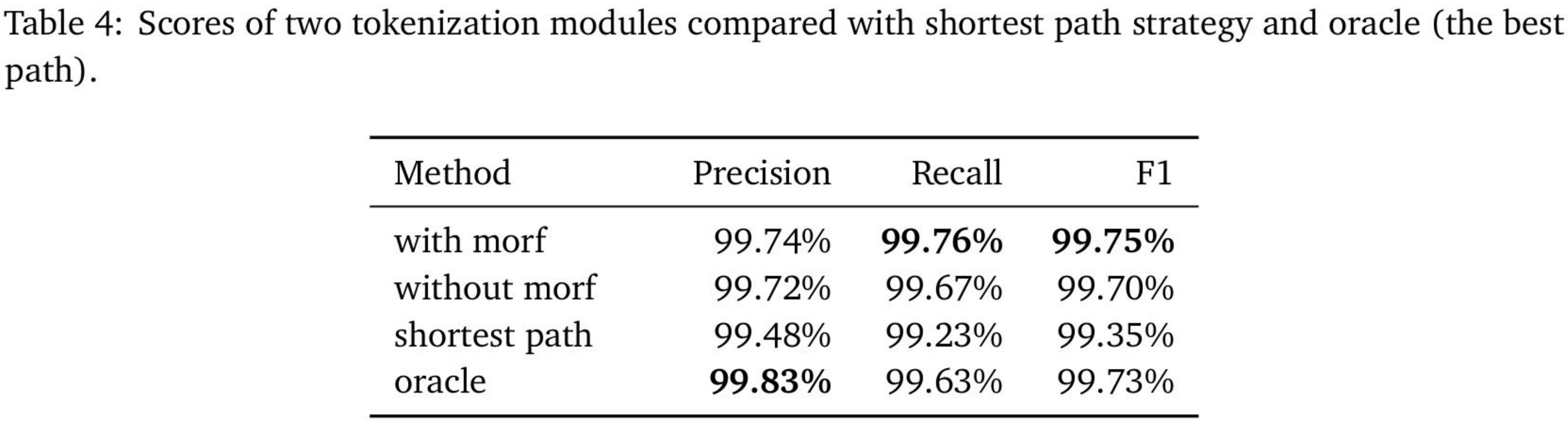


Table 4: Scores of two tokenization modules compared with shortest path strategy and oracle (the best path).

Method	Precision	Recall	F1
with morf	99.74%	**99.76%**	**99.75%**
without morf	99.72%	99.67%	99.70%
shortest path	99.48%	99.23%	99.35%
oracle	**99.83%**	99.63%	99.73%

Results - tagging (PolEval)

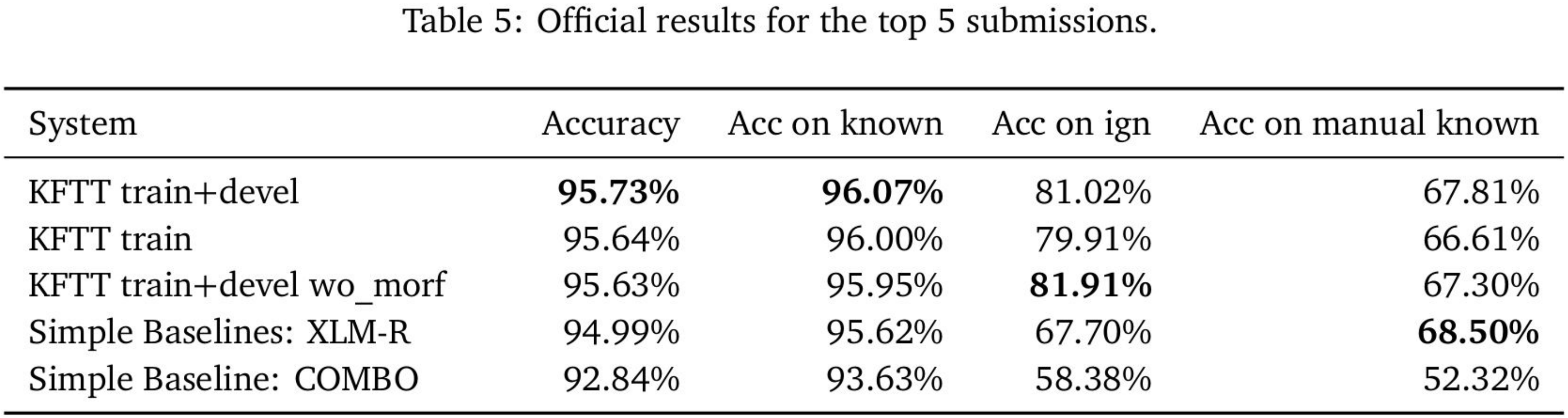


Table 5: Official results for the top 5 submissions.

System	Accuracy	Acc on known	Acc on ign	Acc on manual known
KFTT train+devel	**95.73%**	**96.07%**	81.02%	67.81%
KFTT train	95.64%	96.00%	79.91%	66.61%
KFTT train+devel wo_morf	95.63%	95.95%	**81.91%**	67.30%
Simple Baselines: XLM-R	94.99%	95.62%	67.70%	**68.50%**
Simple Baseline: COMBO	92.84%	93.63%	58.38%	52.32%

Results - tagging

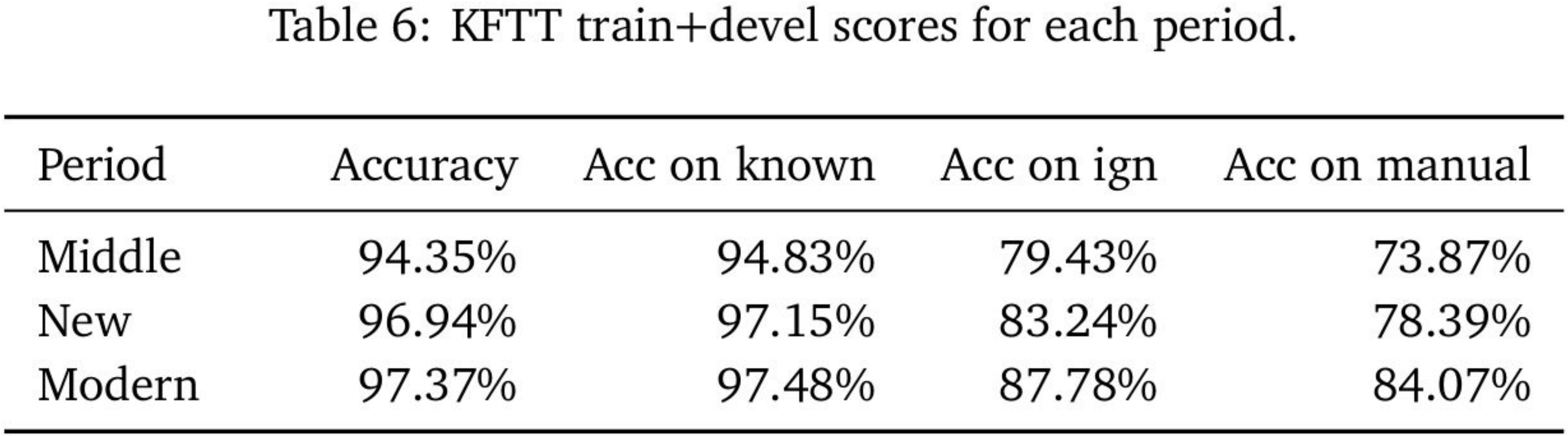


Table 6: KFTT train+devel scores for each period.

Period	Accuracy	Acc on known	Acc on ign	Acc on manual
Middle	94.35%	94.83%	79.43%	73.87%
New	96.94%	97.15%	83.24%	78.39%
Modern	97.37%	97.48%	87.78%	84.07%

For comparison, in 2017 KRNNT on modern texts achieved accuracy 93.72%, on known 94.43%, and on unknown 69.03%.

Source code and models are available at:

https://github.com/kwrobel-nlp/kftt

https://www.linkedin.com/in/wrobelkrzysztof/