Annotating the Tweebank Corpus on Named Entity Recognition and Building NLP Models for Social Media Analysis

Hang Jiang*, Yining Hua*, Doug Beeferman, Deb Roy
MIT Center for Constructive Communication

HARVARD
MEDICAL SCHOOL

LREC 2022
Marseille

Motivation

Annotating named entities in Tweebank enables

- training multi-task learning models in NER, POS tagging, and dependency parsing
- studying linguistic relationship between syntactic labels and named entities in the Twitter domain

Image source: https://www.ebu.ch/groups/social-media-group

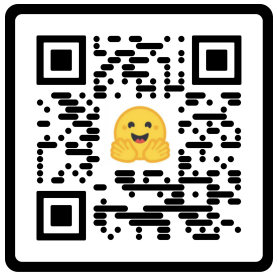
Previous work on *Tweebank*

1. Kong et al. (2014) published Tweebank v1.0
2. Liu et al. (2018) published Tweebank v2.0 (**TB2**)
3. Tweet NLP models on TB2
    a. Tokenization: Twpipe (Liu et al., 2018)
    b. POS tagging: BERTweet (Nguyen et al., 2020)
    c. Dependency parsing: Twpipe (Liu et al., 2018)
4. Tweet NER models on WNUT16 and WNUT17
    a. BERTweet (Nguyen et al., 2020)

Contributions

1. Create ***Tweebank-NER*** benchmark
2. Train and release the ***Twitter-Stanza*** pipeline
3. Compare ***Twitter-Stanza*** against existing models
    - simple neural architecture is effective and suitable for Tweets
4. Train **Transformer-based models**
    - establish a strong baseline on the *Tweebank-NER* benchmark
5. Release our data, models, and code
    - both ***Twitter-Stanza*** and **Hugging Face BERTweet** models

GitHub



Hugging Face

Annotate *Tweebank-NER*

1. CoNLL 2003 guidelines
2. Qualtrics platform + Amazon Mechanical Turk
3. Qualification test
4. Two-stage annotation
    a. 3 annotators annotate each tweet
    b. re-annotate the tweets without consensus

Dataset	Train	Dev	Test
Tweets	1,639	710	1,201
Tokens	24,753	11,742	19,112
Avg. token per tweet	15.1	16.6	15.9
Annotated spans	979	425	750
Annotated tokens	1,484	675	1183
Avg. token per span	1.5	1.6	1.6

Table 1: Annotated corpus statistics.

How is the NER annotation quality?

1. Inter-annotator agreement (IAA)
   - Adopt token-level pairwise F1 score (70.7) calculated without the O label
   - Kappa measure (κ =0.347)
2. MISC (50.9% F1) is the most challenging class for human annotators
3. MISC (47.2%) and ORG (29.2%) are fed for re-annotation

Label	Quantity	F1
PER	777	84.6
LOC	317	74.4
ORG	541	71.9
MISC	519	50.9
Overall	2,154	70.7

Table 2: Number of span annotations per entity type and Inter-annotator agreement scores in pairwise F1.

Methods for NLP Modeling

Models

1. Stanza
2. Hugging Face (BERTweet + Token Classification)
3. spaCy, FLAIR, spaCy-transformers

Questions

1. How do Stanza models perform compared with other NLP frameworks on the core Tweet NLP tasks?
2. How do transformer-based models perform compared with traditional models on these tasks?

HUGGING FACE

spaCy

flair

Performance on *Tweebank-NER*

Performance in *Tweebank-NER*

Main findings

Stanza

1. Stanza NER model (TB2+W17) achieves the best performance among all non-transformer models

2. HuggingFace-BERTweet (TB2+W17) achieves the highest performance (74.35%) on *Tweebank-NER*

3. TB2 and WNUT17 training sets boost the performance

Systems	F1
spaCy (TB2)	52.20
spaCy (TB2+W17)	53.89
FLAIR (TB2)	62.12
FLAIR (TB2+W17)	59.08
HuggingFace-BERTweet (TB2)	73.71
HuggingFace-BERTweet (TB2+W17)	**74.35**
spaCy-BERTweet (TB2)	73.79
spaCy-BERTweet (TB2+W17)	74.15
Stanza (TB2)	60.14
Stanza (TB2+W17)	62.53

Table 3: NER comparison on the TB2 test set in entity-level F1. "TB2" indicates to use the TB2 train set for training. "TB2+W17" indicates to combine TB2 and WNUT17 train sets for training.

Why do we need another Twitter NER dataset?

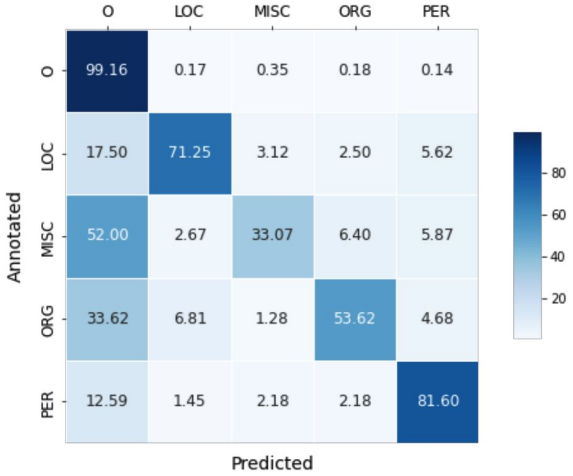
1. The performance of all the models drops significantly if we use the pre-trained model from WNUT17
2. The ***Tweebank-NER*** dataset is still challenging for current NER models
3. Makes ***Tweebank*** a complete dataset for multi-task learning

Training data	TB2	WNUT17	F1 Drop
spaCy	52.20	44.93	7.27↓
FLAIR	62.12	55.11	7.01↓
HgFace-BERTweet	73.71	59.43	14.28↓
spaCy-BERTweet	73.79	60.77	13.02↓
Stanza	60.14	56.40	3.74↓

Table 5: Comparison among NER models trained on TB2 vs. WNUT17 on TB2 test in entity-level F1. "HgFace" stands for "HuggingFace".

NER Error Analysis

1. **PER → O**: when every first letter in a word is capitalized
2. **LOC → O**: when location structure gets too complicated
3. **ORG/MISC → O**: when named entities tend to contain common English verbs



Error type	weet example
PER → O	The 50 % Return Method Billionaire Investor **Warren Buffet** Wishes He Could Use
LOC → O	Getting ready ... @ **Pasco Ephesus Seventh - day Adventist Church**
ORG → O	#bargains #deals 10.27.10 **Guess Who** " American Woman " Guhhh deeeh you !
MISC → O	RT @USER1508 : Do you ever realize **Sounds Live Feels Live** Starts this month and just

Performance on *other Tweebank v2.0* NLP Tasks

Tokenization

Main findings

1. Stanza (TB2) achieves the SOTA
2. Blending TB2 and UD English-EWT for training brings down the tokenization performance slightly

System	F1
Twokenizer	94.6
Stanford CoreNLP	97.3
UDPipe v1.2	97.4
Twpipe	98.3
spaCy (TB2)	98.57
spaCy (TB2+EWT)	95.57
Stanza (TB2)	**98.64**
Stanza (TB2+EWT)	98.59

Table 6: Tokenizer comparison on the TB2 test set. "TB2" indicates to use TB2 for training. "TB2+EWT" indicates to combine TB2 and UD English-EWT for training. Note that the first four results are rounded to one decimal place by Liu et al., (2018).

Lemmatization

Main findings

1. Stanza (TB2) achieves the SOTA
2. Stanza ensemble lemmatizer has both ruled-based dictionary lookup and seq2seq learning
3. TB2 and UD English-EWT training sets hurt the performance

System	F1
NLTK	88.23
spaCy	85.28
Flair (TB2)	96.18
Flair (TB2+EWT)	84.54
Stanza (TB2)	**98.25**
Stanza (TB2+EWT)	85.45

Table 7: Lemmatization results on the TB2 test set. "TB2" is to use TB2 for training. "TB2+EWT" is to combine TB2 and UD English-EWT for training.

POS Tagging

Main findings

1. HuggingFace-BERTweet (TB2+EWT) achieves the SOTA
2. Stanza achieves competitively against greedy Owoputi et al. (2013)
3. TB2 and UD English-EWT training sets boost the performance

System	UPOS
Stanford CoreNLP	90.6
Owoputi et al. (2013) (greedy)	93.7
Owoputi et al. (2013) (CRF)	94.6
Ma and Hovy (2016)	92.5
BERTweet (Nguyen et al., 2020)	95.2
spaCy (TB2)	86.72
spaCy (TB2+EWT)	88.84
FLAIR (TB2)	87.85
FLAIR (TB2+EWT)	88.19
HuggingFace-BERTweet (TB2)	95.21
HuggingFace-BERTweet (TB2+EWT)	**95.38**
spaCy-BERTweet (TB2)	87.61
spaCy-BERTweet (TB2+EWT)	86.31
spaCy-XLM-RoBERTa (TB2)	93.90
spaCy-XLM-RoBERTa (TB2+EWT)	93.75
Stanza (TB2)	93.20
Stanza (TB2+EWT)	93.53

Table 8: POS Tagging comparison in accuracy on the TB2 test set. "TB2" is to use TB2 for training. "TB2+EWT" is to combine TB2 and UD English-EWT for training. Please note that the first five results are rounded to one decimal place by Liu et al., (2018).

Dependency Parsing

Main findings

1. spaCy-XLM-RoBERTa (TB2) achieves the SOTA performance
2. Stanza parser performs competitively against the best non-transformer model – Liu et al. (2018) (Distillation)
3. TB2 and UD English-EWT training sets boost the performance

System	UAS	LAS
Kong et al. (2014)	81.4	76.9
Dozat et al. (2017)	81.8	77.7
Ballesteros et al. (2015)	80.2	75.7
Liu et al. (2018) (Ensemble)	83.4	**79.4**
Liu et al. (2018) (Distillation)	82.1	77.9
spaCy (TB2)	66.93	58.79
spaCy (TB2 + EWT)	72.06	63.84
spaCy-BERTweet (TB2)	76.32	71.72
spaCy-BERTweet (TB2+EWT)	76.18	69.28
spaCy-XLM-RoBERTa (TB2)	**83.82**	**79.39**
spaCy-XLM-RoBERTa (TB2+EWT)	81.02	75.43
Stanza (TB2)	79.28	74.34
Stanza (TB2 + EWT)	82.10	77.60

Table 9: Dependency parsing comparison on the TB2 test set. "TB2" indicates to use TB2 for training. "TB2+EWT" indicates to combine TB2 and UD English-EWT for training. Note that the first six results are rounded to one decimal place by Liu et al., (2018).

Conclusion

1. We introduce four-class named entities to Tweebank V2 - ***Tweebank-NER***
2. We observe great IAA score in pairwise F1 for NER annotation
3. We introduce the ***Twitter-Stanza*** pipeline as a strong Tweet NLP baseline
4. We train and release SOTA **BERTweet models** on TB2
5. We compare Stanza and BERTweet with different NLP frameworks
6. We release our data, models, and code

GitHub

Hugging Face

Future Work

- Develop SOTA Tweet NLP models with multi-task learning
- Design human-in-the-loop methods to identify bad annotation and improve the quality of Tweet NLP datasets

Acknowledgement

- Alan Ritter (Georgia Tech), Yuhui Zhang (Stanford), Zifan Lin (MIT)
- John Bauer (Stanford) and Yijia Liu (University of Washington)
- MIT Center for Constructive Communication (MIT CCC)

HARVARD
MEDICAL SCHOOL

LREC 2022
Marseille