Empirical Evaluation of Character-Based Model on Neural Named-Entity Recognition in Indonesian Conversational Texts

Kemal Kurniawan
Kata Research Team
Kata.ai
Jakarta, Indonesia
kemal@kata.ai

Samuel Louvan
Fondazione Bruno Kessler
University of Trento
Trento, Italy
slouvan@fbk.eu

Abstract

Despite the long history of named-entity recognition (NER) task in the natural language processing community, previous work rarely studied the task on conversational texts. Such texts are challenging because they contain a lot of word variations which increase the number of out-of-vocabulary (OOV) words. The high number of OOV words poses a difficulty for word-based neural models. Meanwhile, there is plenty of evidence to the effectiveness of character-based neural models in mitigating this OOV problem. We report an empirical evaluation of neural sequence labeling models with character embedding to tackle NER task in Indonesian conversational texts. Our experiments show that (1) character models outperform word embedding-only models by up to 4 F_1 points, (2) character models perform better in OOV cases with an improvement of as high as 15 F_1 points, and (3) character models are robust against a very high OOV rate.

1 Introduction

Critical to a conversational agent is the ability to recognize named entities. For example, in a flight booking application, to book a ticket, the agent needs information about the passenger's name, origin, and destination. While named-entity recognition (NER) task has a long-standing history in the natural language processing community, most of the studies have been focused on recognizing entities in well-formed data, such as news articles or biomedical texts. Hence, little is known about the suitability of the available named-entity recognizers for conversational texts. In this work, we tried to shed some light on this direction by evaluating neural sequence labeling models on NER task in Indonesian conversational texts.

Unlike standard NLP corpora, conversational texts are typically noisy and informal. For example, in Indonesian, the word *aku* ("I") can be written as: *aq*, *akuw*, *akuh*, *q*. People also tend to use non-standard words to represent named entities. This creative use of language results in numerous word variations which may increase the number out-of-vocabulary (OOV) words (Baldwin et al., 2013).

The most common approach to handle the OOV problem is by representing each OOV word with a single vector representation (embedding). However, this treatment is not optimal because it ignores the fact that words can share similar morphemes which can be exploited to estimate the OOV word embedding better. Meanwhile, word representation models based on subword units, such as characters or word segments, have been shown to perform well in many NLP tasks such as POS tagging (dos Santos and Zadrozny, 2014; Ling et al., 2015), language modeling (Ling et al., 2015; Kim et al., 2016; Vania and Lopez, 2017), machine translation (Vylomova et al., 2016; Lee et al., 2016; Sennrich et al., 2016), dependency parsing (Ballesteros et al., 2015), and sequence labeling (Rei et al., 2016; Lample et al., 2016). These representations are effective because they can represent OOV words better by leveraging the orthographic similarity among words.

As for Indonesian NER, the earliest work was done by Budi et al. (2005) which relied on a rule-based approach. More recent research mainly used machine learning methods such as conditional random fields (CRF) (Luthfi et al., 2014; Leonandya et al., 2015; Taufik et al., 2016) and support vector machines (Suwarningsih et al., 2014; Aryoyudanta et al., 2016). The most commonly used datasets are news articles (Budi et al., 2005), Wikipedia/DBPedia articles (Luthfi et al., 2014; Leonandya et al., 2015; Aryoyudanta et al., 2016), medical texts (Suwarningsih et al., 2014), and Twitter data (Taufik et al., 2016). To the best of

our knowledge, there has been no work that used neural networks for Indonesian NER nor NER for Indonesian conversational texts.

In this paper, we report the ability of a neural network-based approach for Indonesian NER in conversational data. We employed the neural sequence labeling model of (Rei et al., 2016) and experimented with two word representation models: word-level and character-level. We evaluated all models on relatively large, manually annotated Indonesian conversational texts. We aim to address the following questions:

1) How do the character models perform compared to word embedding-only models on NER in Indonesian conversational texts?
2) How much can we gain in terms of performance from using the character models on OOV cases?
3) How robust (in terms of performance) are the character models on different levels of OOV rates?

Our experiments show that (1) the character models perform really well compared to word embedding-only with an improvement up to 4 F_1 points, (2) we can gain as high as 15 F_1 points on OOV cases by employing character models, and (3) the character models are highly robust against OOV rate as there is no noticeable performance degradation even when the OOV rate approaches 100%.

2 Methodology

We used our own manually annotated datasets collected from users using our chatbot service. There are two datasets: SMALL-TALK and TASK-ORIENTED. SMALL-TALK contains 16K conversational messages from our users having small talk with our chatbot, Jemma.[1] TASK-ORIENTED contains 72K task-oriented imperative messages such as flight booking, food delivery, and so forth obtained from YesBoss service.[2] Thus, TASK-ORIENTED usually has longer texts and more precise entities (e.g., locations) compared to SMALL-TALK. Table 1 shows some example sentences for each dataset. A total of 13 human annotators annotated the two datasets. Unfortunately, we cannot publish the datasets because of proprietary reasons.

SMALL-TALK has 6 entities: DATETIME, EMAIL, GENDER, LOCATION, PERSON, and PHONE. TASK-ORIENTED has 4 entities: EMAIL, LOC, PER, and PHONE. The two datasets have different entity inventory because the two chatbot purposes are different. In SMALL-TALK, we care about personal information such as date of birth, email, or gender to offer personalized content. In TASK-ORIENTED, the tasks usually can be performed by providing minimal personal information. Therefore, some of the entities are not necessary. Table 2 and 3 report some examples of each entity and the number of entities in both datasets respectively. The datasets are tagged using BIO tagging scheme and split into training, development, and testing set. The complete dataset statistics, along with the OOV rate for each split, are shown in Table 4. We define OOV rate as the percentage of word types that do not occur in the training set. As seen in the table, the OOV rate is quite high, especially for SMALL-TALK with more than 50% OOV rate.

As baselines, we used a simple model which memorizes the word-tag assignments on the training data (Nadeau and Sekine, 2007) and a feature-based CRF (Lafferty et al., 2001), as it is a common model for Indonesian NER. We used almost identical features as Taufik et al. (2016) since they experimented on the Twitter dataset which we regarded as the most similar to our conversational texts among other previous work on Indonesian NER. Some features that we did not employ were POS tags, lookup list, and non-standard word list as we did not have POS tags in our data nor access to the lists Taufik et al. (2016) used. For the CRF model, we used an implementation provided by Okazaki (2007)[3].

Neural architectures for sequence labeling are pretty similar. They usually employ a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with CRF as the output layer, and a CNN (Ma and Hovy, 2016) or LSTM (Lample et al., 2016; Rei et al., 2016) composes the character embeddings. Also, we do not try to achieve state-of-the-art results but only are interested whether neural sequence labeling models with character embedding can handle the OOV problem well. Therefore, for the neural models, we just picked the implementation provided in (Rei et al., 2016).[4]

In their implementation, all the LSTMs have

[1] Available at LINE messaging as @jemma.
[2] YesBoss is our hybrid virtual assistant service.
[3] http://www.chokkan.org/software/crfsuite/
[4] https://github.com/marekrei/sequence-labeler

Dataset	Example
SMALL-TALK	sama2 sumatera barat, tapi gue di Pariaman bkn Payakumbuh "also in west sumatera, but I am in pariaman not payakumbuh" rere jem rere, bukan riri. Riri itu siapa deeeh "(it's) rere jem rere, not riri. who's riri?"
TASK-ORIENTED	Bioskop di lippo mall jogja brapa bos? "how much does the movies at lippo mall jogja cost?" Tolong cariin nomor telepon martabak pecenongan kelapa gading, sama tutup jam brp "please find me the phone number for martabak pecenongan kelapa gading, and what time it closes"

Table 1: Example texts from each dataset. SMALL-TALK contains small talk conversations, while TASK-ORIENTED contains task-oriented imperative texts such as flight booking or food delivery. English translations are enclosed in quotes.

Entity	Example
DATETIME	17 agustus 1999, 15februari2001, 180900
EMAIL	dianu#####@yahoo.co.id, b.s#####@gmail.com
GENDER	pria, laki, wanita, cewek
LOCATION/LOC	salatiga, Perumahan Griya Mawar Sembada Indah
PERSON/PER	Yusan Darmaga, Natsumi Aida, valentino rossi
PHONE	085599837###, 0819.90.837.###

Table 2: Some examples of each entity. Some parts are replaced with ### for privacy reasons.

only one layer. Dropout (Srivastava et al., 2014) is used as the regularizer but only applied to the final word embedding as opposed to the LSTM outputs as proposed by Zaremba et al. (2015). The loss function contains not only the log likelihood of the training data and the similarity score but also a language modeling loss, which is not mentioned in (Rei et al., 2016) but discussed in the subsequent work (Rei, 2017). Thus, their implementation essentially does multi-task learning with sequence labeling as the primary task and language modeling as the auxiliary task.

We used an almost identical setting to Rei et al. (2016): words are lowercased, but characters are not, digits are replaced with zeros, singleton words in the training set are converted into unknown tokens, word and character embedding sizes are 300 and 50 respectively. The character embeddings were initialized randomly and learned during training. LSTMs are set to have 200 hidden units, the pre-output layer has an output size of 50, CRF layer is used as the output layer, and early stopping is used with a patience of 7. Some differences are: we did not use any pretrained word embedding, and we used Adam optimization (Kingma and Ba, 2014) with a learning rate of 0.001 and batch size of 16 to reduce GPU memory usage. We decided not to use any pretrained word embedding because to the best of our knowledge, there is no off-the-shelf Indonesian pretrained word embedding that is trained on conversational data. The ones available are usually trained on Wikipedia articles (`fastText`) and we believe it has a very small size of shared vocabulary with conversational texts. We tuned the dropout rate on the development set via grid search, trying multiples of 0.1. We evaluated all of our models using CoNLL evaluation: micro-averaged F_1 score based on exact span matching.

3 Results and discussion

3.1 Performance

Table 5 shows the overall F_1 score on the test set of each dataset. We see that the neural network models beat both baseline models significantly. We also see that the character models consistently outperform the word embedding-only model, where the improvement can be as high as 4 points on SMALL-TALK. An interesting observation is how the improvement is much larger in SMALL-TALK than TASK-ORIENTED. We speculate that this is due to the higher OOV rate SMALL-TALK has, as can be seen in Table 4.

To understand the character model better, we draw the confusion matrix of the word embedding-only and the concatenation model for each dataset in Figure 1. We chose only the concatenation model because both character models are better than the word embedding-only, so we just picked the simplest one.

SMALL-TALK. Both word embedding-only and concatenation model seem to hallucinate PERSON and LOCATION often. This observation is indicated by the high false positive rate of those entities, where 56% of non-entities are recognized as PERSON, and about 30% of non-entities are recognized as LOCATION. Both models appear to confuse PHONE as DATETIME as

SMALL-TALK						TASK-ORIENTED			
DATETIME	EMAIL	GENDER	LOCATION	PERSON	PHONE	EMAIL	LOC	PER	PHONE
90	35	390	4352	3958	83	1707	55614	40624	3186

Table 3: Number of entities in both datasets.

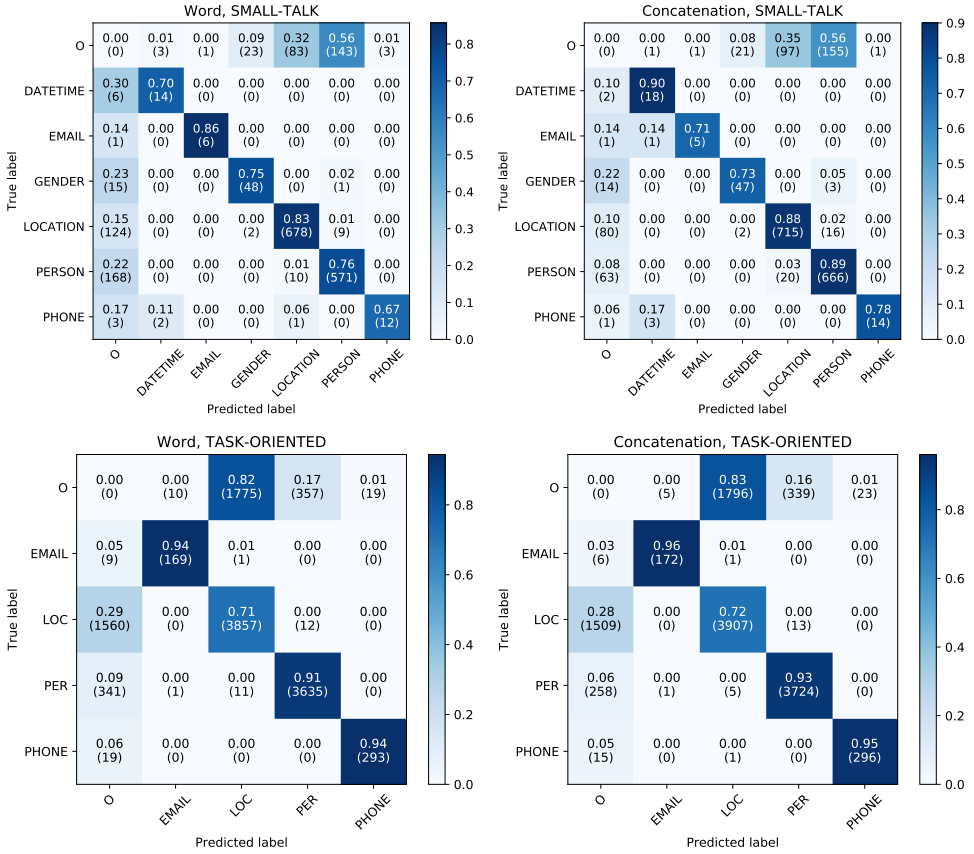

Figure 1: Confusion matrices of the word embedding-only and concatenation model on the test set of each dataset. Top row: SMALL-TALK dataset. Bottom row: TASK-ORIENTED dataset. Left column: word embedding-only model. Right column: concatenation model.

marked by 11% and 17% misclassification rate of the models respectively.

The two models also have some differences. The word embedding-only model has higher false negative than the concatenation model. DATETIME has the highest false negative, where the word embedding-only model incorrectly classified 30% of true entities as non-entity. Turning to the concatenation model, we see how the false negative decreases for almost all entities. DATETIME has the most significant drop of 20% (down from 30% to 10%), followed by PERSON, PHONE, LOCATION, and GENDER.

TASK-ORIENTED. The confusion matrices of the two models are strikingly similar. The models seem to have a hard time dealing with LOC because it often hallucinates the existence of LOC (as indicated by the high false positive rate) and misses genuine LOC entities (as shown by the high false negative rate). Upon closer look, we found that the two models actually can recognize LOC well, but sometimes they partition it into its parts while the gold annotation treats the entity as a single unit. Table 6 shows an example of such case. A long location like *Kantor PKPK lt. 3* is partitioned by the models into *Kantor PKPK* (office name) and *lt. 3* (floor number). The models also partition *Jl Airlangga no. 4-6 Sby* into *Jl Airlangga no. 4-6* (street and building number) and *Sby* (abbreviated city name). We think that this partitioning behavior is reasonable because each part is indeed a location.

		SMALL-TALK	TASK-ORIENTED
L	mean	3.63	14.84
	median	3.00	12.00
	std	2.68	11.50
N	train	10 044	51 120
	dev	3 228	14 354
	test	3 120	7 097
O	dev	57.59	41.39
	test	57.79	32.17

Table 4: Sentence length (L), number of sentences (N), and OOV rate (O) in each dataset. Sentence length is measured by the number of words. OOV rate is the proportion of word types that do not occur in the training split.

Model	SMALL-TALK	TASK-ORIENTED
MEMO	38.03	46.35
CRF	75.50	73.25
WORD	80.96	79.35
CONCAT	84.73	**80.22**
ATTN	**84.97**	79.71

Table 5: F_1 scores on the test set of each dataset. The scores are computed as in CoNLL evaluation. MEMO: memorization baseline. CRF: CRF baseline. WORD, CONCAT, ATTN: Rei et al.'s word embedding-only, concatenation, and attention model respectively.

There is also some amount of false positive on PER, signaling that the models sometimes falsely recognize a non-entity as a person's name. The similarity of the two confusion matrices appears to demonstrate that character embedding only provides a small improvement on the TASK-ORIENTED dataset.

3.2 Performance on OOV entities

Next, we want to understand better how much gain we can get from character models on OOV cases. To answer this question, we ignored entities that do not have any OOV word on the test set and re-evaluated the word embedding-only and concatenation models. Table 7 shows the re-evaluated overall and per-entity F_1 score on the test set of each dataset. We see how the concatenation model consistently outperforms the word embedding-only model for almost all entities on both datasets. On SMALL-TALK dataset, the overall F_1 score gap is as high as 15 points. It is also remarkable that the concatenation model manages to achieve 40 F_1 points for GENDER on SMALL-TALK while the word embedding-only cannot even recognize any GENDER. Therefore,

token	vocab	gold	word	concat
Kantor	kantor	B-LOC	B-LOC	B-LOC
PKPK	UNK	I-LOC	I-LOC	I-LOC
lt	lt	I-LOC	B-LOC	B-LOC
.	.	I-LOC	I-LOC	I-LOC
3	0	I-LOC	I-LOC	I-LOC
,	,	O	O	O
Gedung	gedung	B-LOC	B-LOC	B-LOC
Fak	UNK	I-LOC	I-LOC	I-LOC
.	.	I-LOC	O	I-LOC
Psikologi	psikologi	I-LOC	B-LOC	I-LOC
UNAIR	unair	I-LOC	I-LOC	I-LOC
Kampus	kampus	I-LOC	B-LOC	B-LOC
B	b	I-LOC	I-LOC	B-LOC
.	.	O	O	O
Jl	jl	B-LOC	B-LOC	B-LOC
Airlangga	airlangga	I-LOC	I-LOC	I-LOC
no	no	I-LOC	I-LOC	I-LOC
.	.	I-LOC	I-LOC	I-LOC
4-6	UNK	I-LOC	I-LOC	I-LOC
Sby	sby	I-LOC	B-LOC	B-LOC

Table 6: An example displaying how the word embedding-only (word) and concatenation (concat) models can partition a long location entity into its parts.

in general, this result corroborates our hypothesis that the character model is indeed better at dealing with the OOV problem.

3.3 Impact of OOV rate to model performance

To better understand to what extent the character models can mitigate OOV problem, we evaluated the performance of the models on different OOV rates. We experimented by varying the OOV rate on each dataset and plot the result in Figure 2. Varying the OOV rate can be achieved by changing the minimum frequency threshold for a word to be included in the vocabulary. Words that occur fewer than this threshold in the training set are converted into the special token for OOV words. Thus, increasing this threshold means increasing the OOV rate and vice versa.

From Figure 2, we see that across all datasets, the models which employ character embedding, either by concatenation or attention, consistently outperform the word embedding-only model at almost every threshold level. The performance gap is even more pronounced when the OOV rate is high. Going from left to right, as the OOV rate increases, the character models performance does not seem to degrade much. Remarkably, this is true even when OOV rate is as high as 90%, even approaching 100%, whereas the word embedding-only model already has a significant drop in performance when the OOV rate is just around 70%. This finding confirms that character embedding is useful to mitigate the OOV problem and robust against different OOV rates. We also observe that

Entity	word	concat
DATETIME	50.00	**87.50**
EMAIL	**100.00**	88.89
GENDER	*0.00	**40.00**
LOCATION	51.38	**63.18**
PERSON	68.36	**80.14**
PHONE	0.00	**40.00**
Overall	46.14	**61.75**

Entity	word	concat
EMAIL	95.06	**96.59**
LOC	54.49	**54.74**
PER	73.22	**82.55**
PHONE	***0.00**	**0.00**
Overall	50.05	**54.54**

Table 7: F_1 scores of word embedding-only (word) and concatenation (concat) model on the test set of SMALL-TALK (left) and TASK-ORIENTED (right) but **only** for entities containing at least one OOV word. Entries marked with an asterisk (*) indicate that the model does not recognize any entity at all.

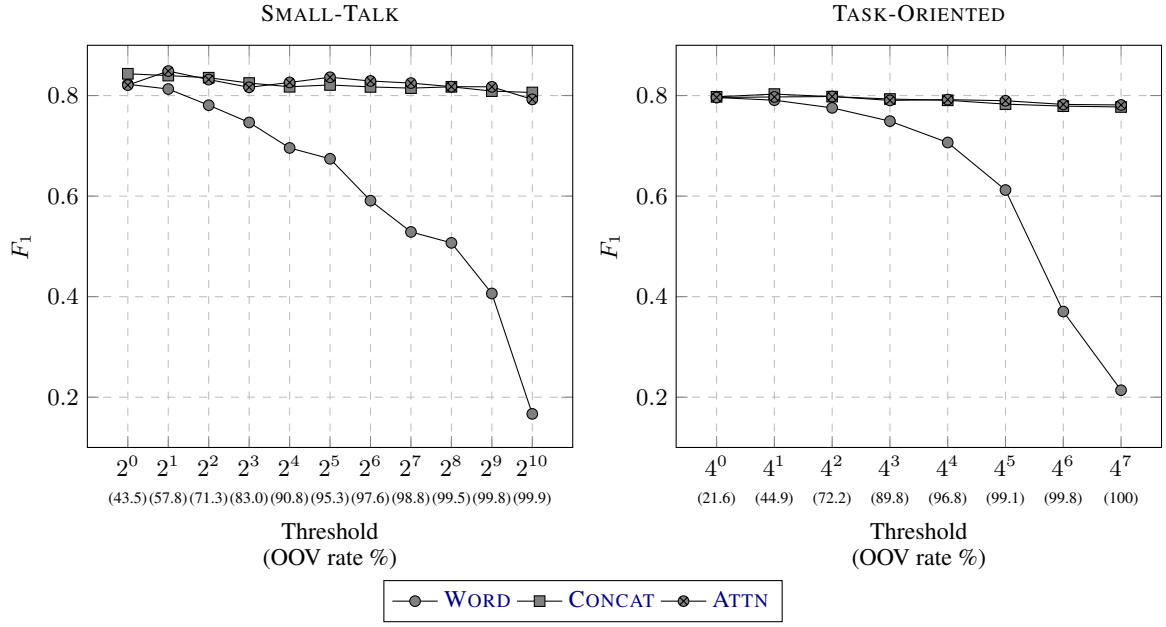

Figure 2: F_1 scores on the test set of each dataset with varying threshold. Words occurring fewer than this threshold in the training set are converted into the special token for OOV words. OOV rate increases as threshold does (from left to right). WORD, CONCAT, and ATTN refers to the word embedding-only, concatenation, and attention model respectively.

there seems no perceptible difference between the concatenation and attention model.

4 Conclusion and future work

We reported an empirical evaluation of neural sequence labeling models by Rei et al. (2016) on NER in Indonesian conversational texts. The neural models, even without character embedding, outperform the CRF baseline, which is a typical model for Indonesian NER. The models employing character embedding have an improvement up to 4 F_1 points compared to the word embedding-only counterpart. We demonstrated that by using character embedding, we could gain improvement as high as 15 F_1 points on entities having OOV words. Further experiments on different OOV rates show that the character models are highly robust against OOV words, as the performance does not seem to degrade even when the OOV rate approaches 100%.

While the character model by Rei et al. (2016) has produced good results, it is still quite slow because of the LSTM used for composing character embeddings. Recent work on sequence labeling by Reimers and Gurevych (2017) showed that replacing LSTM with CNN for composition has no significant performance drop but is faster because unlike LSTM, CNN computation can be parallelized. Using character trigrams as subword units can also be an avenue for future research, as their effectiveness has been shown by Vania and Lopez (2017). Entities like PHONE and EMAIL have quite clear patterns so it might be better to employ a regex-based classifier to recognize such

entities and let the neural network models tag only person and location names.

Acknowledgments

We thank anonymous reviewers for their valuable feedback. We also would like to thank Bagas Swastanto and Fariz Ikhwantri for reviewing the early version of this work. We are also grateful to Muhammad Pratikto, Pria Purnama, and Ahmad Rizqi Meydiarso for their relentless support.

References

B. Aryoyudanta, T. B. Adji, and I. Hidayah. 2016. Semi-supervised learning approach for Indonesian Named Entity Recognition (NER) using co-training algorithm. In *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 7–12.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364, Nagoya, Japan. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Indra Budi, Stéphane Bressan, Gatot Wahyudi, Zainal A. Hasibuan, and Bobby A. A. Nazief. 2005. Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach. In *Discovery Science*, pages 57–69, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1818–1826, Beijing, China. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of the 2016 Conference on Artificial Intelligence (AAAI)*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.

R. A. Leonandya, B. Distiawan, and N. H. Praptono. 2015. A Semi-supervised Algorithm for Indonesian Named Entity Recognition. In *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, pages 45–50.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

A. Luthfi, B. Distiawan, and R. Manurung. 2014. Building an Indonesian named entity recognizer using Wikipedia and DBPedia. In *2014 International Conference on Asian Language Processing (IALP)*, pages 19–22.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. pages 1064–1074. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Naoaki Okazaki. 2007. CRFsuite: A fast implementation of Conditional Random Fields (CRFs).

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

Marek Rei, Gamal K. O. Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, Osaka, Japan.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics. Full experiments report: https://public.ukp.informatik.tu-darmstadt.de/reimers/Optimal_Hyperparameters_for_Deep_LSTM-Networks.pdf.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

W. Suwarningsih, I. Supriana, and A. Purwarianti. 2014. ImNER Indonesian medical named entity recognition. In *2014 2nd International Conference on Technology, Informatics, Management, Engineering Environment*, pages 184–188.

N. Taufik, A. F. Wicaksono, and M. Adriani. 2016. Named entity recognition on Indonesian microblog messages. In *2016 International Conference on Asian Language Processing (IALP)*, pages 358–361.

Clara Vania and Adam Lopez. 2017. From Characters to Words to in Between: Do We Capture Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2016–2027, Vancouver, Canada. Association for Computational Linguistics.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2016. Word representation models for morphologically rich languages in neural machine translation. *arXiv preprint arXiv:1606.04217*.

Wojciech Zaremba, Ilya Sutskever, and Oriols Vinyals. 2015. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.