Deep Affix Features Improve Neural Named Entity Recognizers

Vikas Yadav[†], **Rebecca Sharp**[‡], **Steven Bethard**[†]
[†]School of Information, [‡] Dept. of Computer Science
University of Arizona, Tucson, AZ, USA
{vikasy,bsharp,bethard}@email.arizona.edu

Abstract

We propose a practical model for named entity recognition (NER) that combines word and character-level information with a specific learned representation of the prefixes and suffixes of the word. We apply this approach to multilingual and multi-domain NER and show that it achieves state of the art results on the CoNLL 2002 Spanish and Dutch and CoNLL 2003 German NER datasets, consistently achieving 1.5-2.3 percent over the state of the art without relying on any dictionary features. Additionally, we show improvement on SemEval 2013 task 9.1 DrugNER, achieving state of the art results on the MedLine dataset and the second best results overall (-1.3% from state of the art). We also establish a new benchmark on the I2B2 2010 Clinical NER dataset with 84.70 F-score.

1 Introduction

Named entity recognition (NER), or identifying the specific named entities (eg. person, location, organization etc) in a text, is a precursor to other information extraction tasks such as event extraction. The oldest and perhaps most common approach to NER is based on dictionary lookups, and indeed, when the resources are available, this is very useful (e.g., Uzuner et al., 2011). However, hand-crafting these lexicons is time-consuming and expensive and so these resources are often either unavailable or sparse for many domains and languages.

Neural network (NN) approaches to NER, on the other hand, do not necessitate these resources, and additionally do not require complex feature engineering, which can also be very costly and may not port well from domain to domain and language to language. Commonly, these NN architectures for NER include a learned representation of individual words as well as an encoding of the word's characters. However, neither of these representations makes explicit use of the semantics of sub-word units, i.e., morphemes.

Here we propose a simple neural network architecture that learns a custom representation for affixes, allowing for a richer semantic representation of words and allowing the model to better approximate the meaning of words not seen during training[1]. While a full morphological analysis might bring further benefits, to ease re-implementation we take advantage of the Zipfian distribution of language and focus here on a simple approximation of morphemes as high-frequency prefixes and suffixes. Our approach thus requires no language-specific affix lexicon or morphological tools.

Our contributions are:

1. We propose a simple yet robust extension of current neural NER approaches that allows us to learn a representation for prefixes and suffixes of words. We employ an inexpensive and language-independent method to approximate affixes of a given language using n-gram frequencies. This extension is able to be applied directly to new languages and domains without any additional resource requirements and it allows for a more compositional, and hence richer, representation of words.

2. We demonstrate the utility of including a dedicated representation for affixes. Our model shows as much as a 2.3% F1 improvement over an recurrent neural network model with only words and characters, demonstrating that what our model learns about affixes is complementary to a recurrent layer over characters. We find filtering to high-frequency affixes is essential, as simply using all word-boundary character trigrams degrades performance in some cases.

[1]All code required for reproducibility is available at: https://github.com/vikas95/Pref_Suff_Span_NN

3. We establish a new state-of-the-art for Spanish, Dutch, and German NER, and MedLine drug NER. Additionally, we achieve near state-of-the-art performance in English NER and DrugBank drug NER, despite using no external dictionaries.

2 Related Work

Recent neural network (RNN) state of the art techniques for NER have proposed a basic two-layered RNN architecture, first over characters of a word and second over the words of a sentence (Ma and Hovy, 2016; Lample et al., 2016). Many variants of such approaches have been introduced, e.g., to model multilingual NER (Gillick et al., 2016) or to incorporate transfer-learning (Yang et al., 2016). Such approaches have typically relied on just the words and characters, though Chiu and Nichols (2016) showed that incorporating dictionary and orthography-based features in such neural networks improves English NER. In other domains such as DrugNER, dictionary features are extensively used for NER (Segura Bedmar et al., 2013; Liu et al., 2015), but relying on these resources limits the languages and domains in which an approach can operate, hence we propose a model that does not use external dictionary resources.

Morphological features were highly effective in named entity recognizers before neural networks became the new state-of-the-art. For example, prefix and suffix features were used by several of the original systems submitted to CoNLL 2002 (Sang, 2002; Cucerzan and Yarowsky, 2002) and 2003 (Tjong Kim Sang and De Meulder, 2003) as well as by systems for NER in biomedical texts (Saha et al., 2009). We have used prefix and suffix features by filtering our trigrams based on frequency, which better approximate the true affixes of the language. We show in Section 5 that our filtered set of trigram affixes performs better than simply adding all beginning and ending trigrams. Bian et al. (2014) incorporated both affix and syllable information into their learned word representations. The Fasttext word embeddings (Bojanowski et al., 2017) represent each word as a bag of n-grams and thus incorporate sub-word information. Here, we provide explicit representation for only the high-frequency n-grams and learn a task-specific semantic representation of them. We show in Section 5 that including all n-grams reduces performance.

Other sub-word units, such as phonemes (from Epitran[2] - a tool for transliterating orthographic text as International Phonetic Alphabet), have also been found to be useful for NER (Bharadwaj et al., 2016). Tkachenko and Simanovsky (2012) explored contributions of various features, including affixes, on the CoNLL 2003 dataset. Additionally, morpheme dictionaries have been effective in developing features for NER tasks in languages like Japanese (Sasano and Kurohashi, 2008), Turkish (Yeniterzi, 2011), Chinese (Gao et al., 2005), and Arabic (Maloney and Niv, 1998). However, such morphological features have not yet been integrated into the new neural network models for NER.

3 Approach

We consider affixes at the beginnings and ends of words as sub-word features for NER. Our base model is similar to Lample et al. (2016) where we apply an long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) layer over the characters of a word and then concatenate the output with a word embedding to create a word representation that combines both character-level and word-level information. Then, another layer of LSTM is applied over these word representations to make word-by-word predictions at the sentence level. Our proposed model augments this Lample et al. (2016) architecture with a learned representation of the n-gram prefixes and suffixes of each word.

3.1 Collecting Approximate Affixes

We consider all n-gram prefixes and suffixes of words in our training corpus, and select only those whose frequency is above a threshold, T, as frequent prefixes and suffixes should be more likely to behave like true morphemes of a language.To determine the n-gram size, n, and the frequency threshold, T, we experimented with various combinations of $n = 2, 3, 4$ and $T = 10, 15, 20, 25, 50, 75, 100, 150, 200$ by filtering affixes accordingly and evaluating our model (described below) on the CoNLL 2002 and CoNLL 2003 validation data. The best and consistent parameter setting over all 4 languages was $n = 3$ (three character affixes) and $T = 50$ (affixes that occurred at least 50 times in the training data). We have used $n = 3$ and $T = 10$ for DrugNER after getting best performance with this threshold on val-

[2] https://pypi.org/project/epitran/0.4/

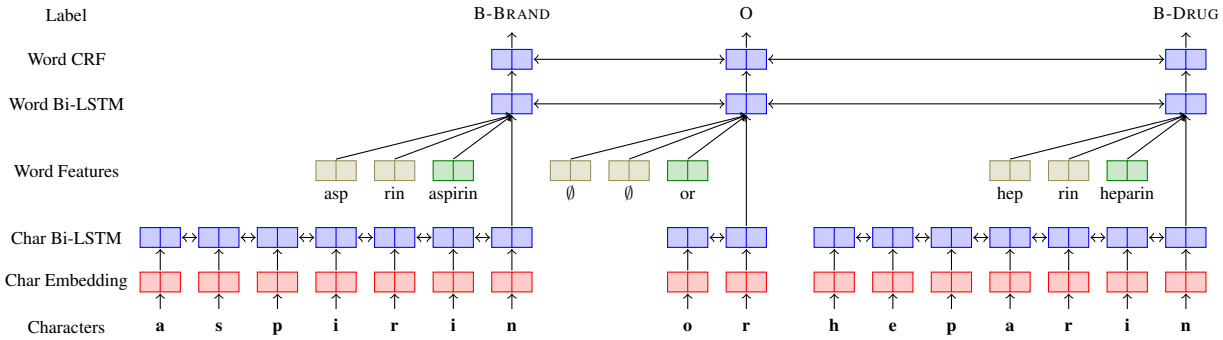

Figure 1: Architecture of our approach. We concatenate a learned representation for our approximated affixes (shown in brown) to a Bi-LSTM encoding of the characters (in blue) and the learned representation of the word itself (in green). This is then passed through another Bi-LSTM and CRF to produce the named entity tags.

idation data and we have used $T = 20$ for I2B2 NER dataset.

3.2 Model and Hyper-parameters

Our proposed model, shown in Figure 1, has separate embeddings for characters, prefixes, suffixes, and words. First, a character embedding maps each of the characters of a word to a dense vector. Then a bidirectional-LSTM (Bi-LSTM) layer is passed over the character embeddings to produce a single vector for each word. The output of this Bi-LSTM layer is concatenated with embeddings for the prefix, suffix, and the word itself, and this concatenation is the final representation of the word. Then the representations of each word in the sentence are passed through another Bi-LSTM layer, followed by a conditional random field (CRF) layer, to produce the begin-inside-outside (BIO) named entity tags.

We randomly initialized character, prefix and suffix affix embeddings. We used Fasttext 300-dimension word embeddings (Bojanowski et al., 2017) for Spanish, Dutch CoNLL 2002 and German language CoNLL 2003. We experimented with 300-dimension Fasttext embeddings and 100-dimension Glove embeddings for CoNLL 2003 English data and saw no appreciable differences ($\pm$ 0.2%). Thus, we report scores with 100-dimension Glove embeddings due to the reduced training time and fewer parameters. We used 300 dimension Pubmed word embeddings (Pyysalo et al., 2013) for DrugNER and I2B2 clinical NER. Across all evaluations in the Section 4, we use the same hyper-parameter settings: Character embedding size = 50; prefix embedding size = 30; suffix embedding size = 30; hidden size for LSTM layer over characters = 25; hidden size for LSTM layer over [prefix, suffix, word, LSTM(characters)] = 50; maximum number of epochs = 200; early stopping = 30 (i.e., if no improvement in 30 epochs, stop); dropout value = 0.55, applied after concatenating character LSTM representation, word embedding and affix embedding; learning rate (LR) = 0.15; LR decay rate= 0.99; optimizer = SGD; and batch size = 100 (for all datasets except Dutch = 80).

4 Experiments

We evaluate our model across multiple languages and domains.

4.1 Multilingual Datasets

To evaluate on the CoNLL 2002 and 2003 test sets, we trained our model on the combined training + validation data with the general hyper-parameter set from Section 3.2. Since on the validation data, the majority of our models terminated their training between 100 and 150 epochs, we report two models trained on the combined training + validation data: one after 100 epochs, and one after 150 epochs.

We evaluated our model with all the languages in CoNLL 2002 and 2003, as reported in Table 1. Our model achieved state of the art performance on Spanish CoNLL 2002 (Sang, 2002), outperforming Yang et al. (2016) by 1.49%, on Dutch CoNLL 2002, outperforming Yang et al. (2016) by 2.35%, and on German CoNLL 2003, outperforming Lample et al. (2016) by 0.25%. Our reimplementation of Lample et al. (2016) using Fasttext word embedding (Dutch) could also achieve state of the art results on Dutch CoNLL 2002 dataset. This demonstrates the utility of our affix approach, despite its simplicity.

	Dict	ES	NL	EN	DE
Gillick et al. (2016) – Byte-to-Span (BTS)	No	82.95	82.84	86.50	76.22
Yang et al. (2016)	No	85.77	85.19	91.26	-
Luo et al. (2015)	Yes	-	-	91.20	-
Chiu and Nichols (2016)	Yes	-	-	**91.62 (±0.33)**	-
Ma and Hovy (2016)	No	-	-	91.21	-
Lample et al. (2016)	No	85.75	81.74	90.94	78.76
Our base model (100 Epochs)	No	85.34	85.27	90.24	78.44
Our model (with Affixes) (100 Epochs)	No	86.92	87.50	90.69	78.56
Our model (with Affixes) (150 Epochs)	No	**87.26**	**87.54**	90.86	**79.01**

Table 1: Performance of our model (with and without affixes), using general set of hyper-parameters and previous work on four datasets: CoNLL 2002 Spanish (ES), CoNLL 2002 Dutch (NL), CoNLL 2003 English (EN), and CoNLL 2003 German (DE). Dict indicates whether or not the approach makes use of dictionary lookups.

Model	Dict	ML (80.10%)			DB (19.90%)			Both datasets		
		P	R	F1	P	R	F1	P	R	F1
Rocktäschel et al. (2013)	Yes	60.7	55.8	58.10	88.10	87.5	87.80	73.40	69.80	71.50
Liu et al. (2015) (baseline)	No	-	-	-	-	-	-	78.41	67.78	72.71
Liu et al. (2015) (MedLine emb.)	No	-	-	-	-	-	-	82.70	69.68	75.63
Our model (with affixes)	No	74	64	**69**	89	86	87	81	74	77
Liu et al. (2015) (state of the art)	Yes	78.77	60.21	68.25	90.60	88.82	**89.70**	84.75	72.89	**78.37**

Table 2: DrugNER results with official evaluation script on test dataset consisting of MedLine (ML) (80.10% of the total test data) and DrugBank (DB) test data (19.90 % of the total test data). We report precision (P), recall (R), and F1-score.

4.2 Clinical and Drug NER

To prove the effectiveness of our proposed model in multiple domains, we also evaluated our model on the SemEval 2013 task 9.1 DrugNER dataset (Segura Bedmar et al., 2013) and the I2B2 clinical NER dataset (Uzuner et al., 2011) .

We first converted these datasets into CoNLL BIO format and then evaluated the performance with CoNLL script. We have also evaluated DrugNER performance with the official evaluation script (Segura Bedmar et al., 2013)[3] after converting it to the required format. These results are given in Table 2. The SemEval 2013 task 9.1 DrugNER dataset is composed of two parts: the MedLine test data which consists of 520 sentences and 382 entities, and the DrugBank test data which consists of 145 sentences and 303 entities. We outperform Liu et al. (2015) by 0.75% and Rocktäschel et al. (2013) by 10.90% on MedLine test dataset. On the overall dataset, we outperform Liu et al.'s dictionary-free model and Rocktäschel et al. by at least 6.50 percent. Again, this shows the benefit from allowing the model to learn a representation of affixes as well as of words and characters. Overall, we achieved the second best result after Liu et al. (2015) but get state of the art results on MedLine test dataset which is 80.10% of the total test data.

For fair comparison with previous work (Unanue et al., 2017) which has re-implemented Lample et al. (2016) model, we tested our model on BIO converted dataset used by Unanue et al. (2017). The results are summarized in table 3.

On the I2B2 NER dataset (Uzuner et al., 2011; Unanue et al., 2017) in the BIO format, we evaluated our approach using the CoNLL 2003 evaluation script. Our final model achieves 84.70 F-score, a gain of 3.68% as compared to the base model without affixes (81.02%) and a gain of 0.67 % over the model of Unanue et al. (2017). For fair comparison with Unanue et al. (2017), we provide results on the I2B2 NER dataset in BIO format evaluated with the CoNLL 2003 evaluation script in Table 4.

[3]The official evaluation script available on the SemEval 2013 website outputs only whole numbers, despite the shared task reporting results to 2 decimal places.

Model	drug	brand	group	drug_n	ML	drug	brand	group	drug_n	DB	Both
Unanue et al. (2017)	75.57	28.57	64.37	37.19	60.66	91.83	87.27	84.67	0	88.38	-
BASE	72	41.67	75.86	4.88	60.86	89.92	79.12	86.13	0	86.52	72.31
BASE+Affix(10)	79.25	44.44	85.39	32.73	69.71	92.09	86.60	87.41	20	88.93	78.39

Table 3: DrugNER results on test data using CoNLL evaluation script. ML indicates the results for MedLine test data and DB indicates results for DrugBank test data. We have reported F1 scores for each entity type in MedLine, DrugBank and overall dataset (Both). The last column (Both) provides performance on the the combined dataset.

Model	Problem			Test			Treatment		
	P	R	F1	P	R	F1	P	R	F1
Unanue et al. (2017)	81.29	83.62	82.44	84.74	85.01	84.87	83.36	83.55	83.46
Base Model	82.45	77.88	80.10	87.24	77.96	82.34	85.53	76.97	81.02
Base+Affix(20)	84.35	84.27	84.31	87.37	84.34	85.82	85.73	82.58	84.13

Table 4: Performance on I2B2 2010 NER (Uzuner et al., 2011) test data [5] using CoNLL evaluation script. We have reported precision (P), recall (R), and F1-score.

5 Analysis

To better understand the performance of our model, we conducted several analyses on the English CoNLL 2003 dataset.

To determine if the performance gains were truly due to the affix embeddings, and not simply due to having more model parameters, we re-ran our base model (without affixes), increasing the character embeddings from 25 to 55 to match the increase of 30 of our affix embeddings. This model's F-score (90.28%) was similar to the original base model (90.24%), and was more than a half a point below our model with affixes (90.86%).

To determine the contribution of filtering our affixes based on frequency (as compared to simply using all word-boundary n-grams) we ran our model with the full set of affixes found in training. The performance without filtering (89.87% F1) was even lower than the base model without affixes (90.24% F1), which demonstrates that filtering based on frequency is beneficial for affix selection.

6 Conclusion

Our results across multiple languages and domains show that sub-word features such as prefixes and suffixes are complementary to character and word-level information. Our straight-forward and language-independent approach shows performance gains compared to other neural systems for NER, achieving a new state of the art on Spanish, Dutch, and German NER as well as the MedLine portion of DrugNER, despite our lack of dictionary resources. Additionally, we also achieve 3.67% improvement in the I2B2 clinical NER dataset which points towards potential applications in biomedical NER. While our model proposes a very simple idea of using filtered affixes as an approximation of morphemes, we suggest there are further gains to be had with better methods for deriving true morphemes (e.g., the supervised neural model of Luong et al., 2013). We leave this exploration to future work.

References

Akash Bharadwaj, David R. Mortensen, Chris Dyer, and Carlos de Juan Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *EMNLP*.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 132–148.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370.

Silviu Cucerzan and David Yarowsky. 2002. Language independent ner using a unified model of internal and contextual evidence. In *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, pages 1–4.

Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese word segmentation and

named entity recognition: A pragmatic approach. *Computational Linguistics* 31(4):531–574.

Daniel Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *HLT-NAACL*.

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

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*. Association for Computational Linguistics, pages 260–270.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2015. Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. *Information* 6(4):848–865.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 879–888.

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*. pages 104–113.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany, pages 1064–1074.

John Maloney and Michael Niv. 1998. Tagarab: a fast, accurate arabic name recognizer using high-precision morphological analysis. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*. Association for Computational Linguistics, pages 8–15.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Languages in Biology and medicine*. LBM.

Tim Rocktäschel, Torsten Huber, Michael Weidlich, and Ulf Leser. 2013. Wbi-ner: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In *SemEval@ NAACL-HLT*. pages 356–363.

Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. 2009. Feature selection techniques for maximum entropy based biomedical named entity recognition. *Journal of Biomedical Informatics* 42(5):905 – 911. Biomedical Natural Language Processing.

Erik F Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: language-independent named entity recognition, proceedings of the 6th conference on natural language learning. *August* 31:1–4.

Ryohei Sasano and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *IJCNLP*. pages 607–612.

Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pages 142–147.

Maksim Tkachenko and Andrey Simanovsky. 2012. Named entity recognition: Exploring features. In *KONVENS*. pages 118–127.

Iñigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. 2017. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. *Journal of biomedical informatics* 76:102–109.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association* 18(5):552–556.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2016. Transfer learning for sequence tagging with hierarchical recurrent networks. *CoRR* abs/1703.06345.

Reyyan Yeniterzi. 2011. Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*. Association for Computational Linguistics, pages 105–110.