Using Transfer Learning in Part-Of-Speech Tagging of English Tweets

Sara Meftah*, **Nasredine Semmar***, **Othman Zennaki***, **Fatiha Sadat**+

*CEA, LIST, Vision and Content Engineering Laboratory
CEA Saclay Nano-INNOV, F-91191, Gif-sur-Yvette, France
{sara.meftah, nasredine.semmar, othman.zennaki}@cea.fr
+Université du Québec à Montréal, UQÀM
201 Président Kennedy Avenue H2X 3Y7, Montréal, Canada
sadat.fatiha@uqam.ca

Abstract

This paper presents a Part-Of-Speech (POS) tagger for English tweets which uses an end-to-end Deep Neural Networks (DNN) model requiring no feature engineering or data pre-processing. Our neural model benefits from both word and character level representations. Character level word representations are learned during the training of the model through a Convolutional Neural Network (CNN). For word level representations, we concatenate several embeddings to accurately capture the word semantics. We evaluate our model on two publicly available small data sets of English tweets. We propose an approach to handle the problem of annotated data availability and demonstrate that transfer learning improves the performance of POS tagging.

1. Introduction

Part-of-Speech (POS) tagging is one of the basic and indispensable tasks in Natural Language Processing (NLP). Most traditional high performance POS tagging models are linear statistical models. These models rely heavily on hand-crafted features and task specific external resources. However, such task-specific knowledge is costly to develop and makes POS tagging models difficult to adapt to new tasks or domains.

The past few years have witnessed the great success of the application of deep neural networks in end-to-end manner for Natural Language Processing (NLP). Most of proposed neural models for sequence labeling, including POS tagging use Recurrent Neural Networks (RNNs) and its variants (Long Short-Term Memory networks - LSTMs and Gated Recurrent Units - GRUs), and Covolutionnal Neural Networks (CNNs) for character-level representations. Indeed, previous studies (Jozefowicz et al., 2016) have shown that CNNs represent an effective approach to extract morphological information (root, prefix, suffix, etc.) from words and encode it into neural representations, especially for morphological rich texts like Twitter data (Chiu and Nichols, 2015; Ma and Hovy, 2016).

In fact, the current challenge is not POS tagging of well-structured texts with huge amounts of labeled corpora, since the actual accuracy of POS taggers trained from treebanks in the newswire domain, such as the Wall Street Journal (WSJ) corpus of the Penn Treebank (Marcus et al., 1993) is close to human level, thanks to deep learning techniques trained on large annotated data-sets (97.64% accuracy by (Choi, 2016)). However, achieving human-level accuracy for POS tagging on user generated texts, especially conversational texts (Twitter, Web blogs, SMS texts, etc.) is much harder. This is due to the conversational nature of the text, the lack of conventional orthography, the noise, linguistic errors and the idiosyncratic style. Also, Twitter poses an additional issue by imposing 140 characters limit for each Tweet.

The application of models purely trained on well-structured corpora such as WSJ falls to work on noisy text such as twitter. As illustrated in (Gimpel et al., 2011), the accuracy of the Stanford POS tagger (Toutanova et al., 2003) trained on WSJ falls from 97% on standard English to 85% accuracy on Twitter. The main reason for this drop in accuracy is that Twitter contains lot of Out-Of-Vocabulary (OOV) words compared to standard text. In addition, NLP's deep neural network (DNN) models with high performance often require huge volumes of annotated data to produce powerful models and prevent over-fitting. Hence, the construction of a DNN model for Twitter data needs huge amounts of annotated tweets with POS labels to produce high performance, however, available data-sets are very small.

There are two principal state-of-the-art works for English Tweets POS tagging, both based on hand-crafted features. Derczynski et al., 2013) used a small data-set of annotated tweets. The model is based on hidden Markov Models and a set of normalization rules, external dictionaries and lexical features. They achieve an accuracy of 88.69%. Furthermore, they achieve 90.54% token accuracy using supplementary 1.5M training tokens annotated by vote-constrained bootstrapping. Owoputi et al., 2013) used another small data-set of annotated tweets. The model is based on First-order maximum entropy Markov model (MEMM), engineered features like brown clustering and lexical features. They achieve an accuracy of 93.2%.

Recently a Neural Network model (TPANN) for English Tweets POS tagging was proposed by Gui et al., 2017), they used Adversarial Neural Networks to leverage huge amounts of unlabeled Tweets and labeled out-of-domain data (WSJ) [1]. TPANN achieves high performances compared to the former works. However, the model proposed in (Gui et al., 2017) requires that labeled in-domain-data and labeled out-of-domain data share the same tag-set (a mapping is necessary in case of tag-sets mismatch). Which makes the model difficult to adapt to new tasks or

[1] The adversarial discriminator helps to find an invariant representation of in-domain and out-of-domain data.

new domains, where the mapping between tag-sets is not possible.

In the past few years transfer learning has been successfully applied in neural speech processing and machine translation (Zoph et al., 2016). It consists in performing a task on a low-resource target data-set using features learned from a high-resource source data-set (Pan and Yang, 2010). Two studies have been recently performed on transfer learning for DNN based models in sequence labeling: (Yang et al., 2017) and (Lee et al., 2017) for named entity recognition.

In this work, we use transfer learning in a simple DNN model for POS tagging of English Tweets. In order to handle the constraints of the high frequency of OOV words, the very limited size of the POS labeled in-domain data and the absence of parallel corpora between English tweets and the standard English texts, we propose:

- An end-to-end and feature-engineering-free deep neural model for English Tweets POS tagging, with a small annotated corpus.

- To combine different input representations in order to handle OOV words issue.

- To exploit huge amounts of available out-of-domain annotated corpora using transfer learning, we demonstrate that transferring a DNN model trained on out-of-domain data (Ex. Standard English) to another small data set (English tweets) improves the state-of-the-art results despite domain and tag-sets mismatch.

The remainder of the paper is organized as follows. We first describe in section 2 our neural model, then, we present in section 3 the experimental setup. In Section 4, we report and discuss the results. Finally, the conclusion and future work are presented in section 5.

2. System description

Our model is based on deep hierarchical Gated Recurrent Units (GRU), a type of Recurrent Neural Networks (RNN), and utilizes both word-level and character-level embeddings. Transfer learning is applied in order to leverage out-of-domain data even if source and target data sets do not share the same tag-set (more details about Transfer learning approach are reported in the section 3.4.).

2.1. Deep Neural Network model

Our neural network model comprises three major components, Figure 1 shows an overview of the architecture.

2.1.1. Inputs Representation

In order to preserve both semantic and syntactic information of words, we join character-level and word-level embeddings to get a combined embedding. Hence, each word in the input tweet is represented by a combination of two vectors:

1. A vector representation of individual words: Instead of using a single pre-trained word embedding model as the final word-level representation, we initialize word-level embedding with a concatenation of different pretrained words embedding (Section 3.3.) to accurately capture the word's semantics. Indeed, our experiments show that this method can produce better performances (Section 4.2.1.).

2. Character-level word embedding: In order to deal with OOV words, we represent each word with a vector that contains morphological information, generated using a Convolutional Neural Network (CNN) (the same architecture used by Ma and Hovy, 2016), except that we add an embedding layer before the convolutional layer).

2.1.2. Gated Recurrent Units Layer

Word vectors (the combination between character level embedding and word level embedding CNN) are fed into a 100 dimension GRU layer.

2.1.3. Fully-connected Layer and Softmax Layer

The output of the GRU at each time-step is fed through a 80 dimension linear (fully connected) layer followed by a softmax layer to decode it into a distribution of log-probabilities for each POS tag.

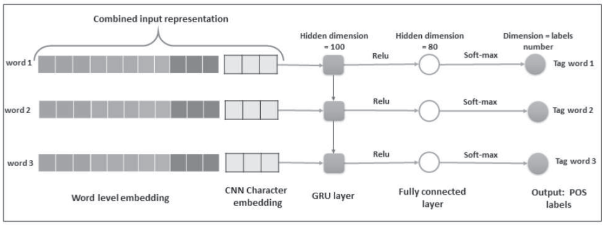

Figure 1: Overall system design. First, the system embeds each word of the tweet into two representations: character level representation using a CNN network and a word level representation by combining different pre-trained models. Then, the two representations are combined and fed into a GRU layer, the resulting vector is fed to a fully connected layer and finally a softmax layer to perform POS tagging.

3. Experimentation

3.1. Data sets

For experiments, we use the two POS-labeled Tweets data-sets that are currently publicly available:

- The T-PoS corpus of 787 hand-annotated English tweets (15K tokens) introduced by (Ritter et al., 2011), which uses the same tag-set as Penn Tree-bank (PTB) tag-set (Marcus et al., 1993), plus four Twitter special tags: *URL* for URLs, *HT* for hashtags, *USR* for username mentions and *RT* for retweet signifier (40 tags in total).

- The ARK corpus was published on two parts, the first, Oct27 of 1827 hand-annotated English tweets (39K tokens), published in (Gimpel et al., 2011) and the second, Daily547 of 547 tweets published in (Owoputi et al., 2013), using a novel and coarse grained tag-set (25 tags). For example, its V tag corresponds to any verb, conflating PTB's VB, VBD, VBG, VBN, VBP, VBZ, and MD tags.

Nevertheless, huge amounts of near-genre annotated corpora are also available:

- Standard English: The WSJ part of the PTB annotated with part-of-speech tags. The PTB tag-set includes 36 main tags and an additional 12 tags covering items such as punctuation.
- Tweet-like genre data: The NPS IRC Chat Corpus (annotated with part-of-speech tags) (Forsythand and Martell, 2007) consists of 10,567 posts gathered from various online chat services.

3.2. Baselines

We compare our system's performances to three models:

- Derczynski et al., 2013) used T-PoS data, by splitting it 70:15:15 into training, development and evaluation sets named T-train, T-dev and T-eval. For training, they used T-train (2.3K tokens) and also 50K tokens from the Wall Street Journal part of the Penn Treebank and 32K tokens from the NPS IRC corpus. We use the same data splits for our experiments on T-POS.
- Owoputi et al., 2013) used the Ark corpus oct27 as training and development data data and Daily547 as an evaluation data. We split the oct27 data set into training-set and development-set (70:30) (data splits portions are not mentioned) and Daily547 as validation set.
- Gui et al., 2017) performed experiments on both T-POS and ARK data-sets. For training, they leverage 1,17M token from unlabeled Tweets and more than 1,17M from labeled WSJ. In order to use WSJ labeled data in experiments on ARK data set, they performed a mapping between PTB and ARK tag-sets.

3.3. Word embedding

We experimented an initialization of words embedding with different sets of published pre-trained vectors. Initializations are computed by a look-up table of each of pre-trained model, all words are lower-cased before passing through the look up table for converting to their corresponding vectors.

1. Word2vec (Mikolov et al., 2013), which learns word vector representations by attempting to predict context words around an input word, trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words.[2]
2. FastText (Bojanowski et al., 2016), which is very similar to Word2vec (Using SkipGram) but it also uses sub-word information in the prediction model. FastText Facebook embedding is trained on Wikipedia for 294 languages and contains 300 dimensional words vectors.[3]
3. Glove (Pennington et al., 2014) is a model based on global word-word co-occurrence statistics, we experimented two Glove's models, the first, which we name "Glove", trained on 42 billions from a web crowling, contains 300 dimensional vectors for 1.9M words. And the second, which we name "Glove-Twitter", trained on 2 billion tweets, contains 200 vectors for 1.2M words.[4]

We have also done experiments with randomly initialized embeddings of 300 dimensions.

3.4. Transfer learning

In our case transfer learning helps to leverage the vast amounts of labeled data we have in structured English (WSJ) to improve English tweets POS tagging. Furthermore, we use it to cope with the problem of the mismatch between the tag-sets of available POS-labeled Tweets.

Since the two available data sets (T-POS and Ark) do not share the same tag-set, we perform experiments separately. The first experiment was to evaluate the performance of the DNN trained on the available annotated tweets (T-train for T-POS and oct27 for ARK) without any extra knowledge. Then, for experiments on T-POS, we augment in-domain data with out-of-domain annotated data (WSJ + IRC) and train the DNN jointly on T-train, WSJ and IRC[5].

Finally, we use transfer learning approach by training the parent model on a high-resource out-of-domain data, then use the parameters (weights) of this model to initialize the child model which is further trained (fine-tuned) on a low-resource in-domain-data (Tweets), rather than starting from a random position. For experiments on ARK, the parent model is trained on T-POS+WSJ+IRC, and trained on WSJ+IRC for experiments on T-POS (We show the most effective layers for transfer for both data-sets in 4.2.).

4. Results and discussion

In this section, we firstly report best performances of our model compared to the baselines described in 3.2. Then, we show the importance of transfer learning and pre-trained word embedding.

4.1. Performances Evaluation

In table 1, we show our system's performances compared to state-of-the-art results. The first part of the table shows (Derczynski et al., 2013), (Owoputi et al., 2013) and (Gui et al., 2017) results. The second part shows the results of our model trained only on in-domain-data (T-train for T-POS and oct27 for ARK) and then results by

[2] code.google.com/archive/p/word2vec/

[3] github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[4] nlp.stanford.edu/projects/glove/

[5] For ARK data-set, it is not possible to perform a joint-training with out-of-domain data because of the tag-set mismatch

Data sets	T-POS (T-eval)		ARK (Daily547)
Methods	Token acc. (%)	Sentence acc. (%)	Token acc. (%)
(Derczynski et al., 2013)	88.69	20.34	–
(Owoputi et al., 2013)	90.40	–	**93.20**
(Gui et al., 2017)	**90.92**	–	92.8
In-domain-data training	88.13	21.03	90.59
Jointly training	89.01	22.04	–
Transfer learning	**90.46**	**23.01**	**91.66**

Table 1: Performances of different methods on T-eval (Validation set of T-POS) and Daily547 (Validation set of ARK). ”In-domain-data training” refers to training our DNN model only on Twitter data. ”Jointly training” refers to the model trained jointly on T-train, WSJ and IRC for T-POS experiments. ”Transfer learning” refers to the model trained firstly on (WSJ + IRC) and then fine-tuned on T-Train for T-POS experiments, and on (WSJ + IRC + T-POS) and then fine-tuned on oct27 for ARK experiments.

augmenting training set with WSJ and IRC data for experiments on T-POS (same training set as (Derczynski et al., 2013))[6]. The third part shows the results using transfer learning.

Comparing the two methods of using out-of-domain data, we can observe that transfer learning method can achieve better performance than the jointly training method (almost 1.3% higher token-accuracy for T-POS).

Our transfer learning method outperforms state-of-the-art approaches (Derczynski et al., 2013) and (Owoputi et al., 2013) on T-POS experiments, however it performs worse than (Owoputi et al., 2013) on ARK-data set. The reason is that our model is end-to-end and the most of errors in our system were caused by hashtags (Our model accuracy on hashtags: 45%) and proper nouns (PN) (Our model accuracy on PN: 55%) which was resolved in (Owoputi et al., 2013) by adding external knowledge (a list of named entities) and rules to detect hashtags.

We can observe that (Gui et al., 2017) achieves better performances than our model in both data sets, an effective model (Adversarial Neural Network) was used in their work with huge amounts of unlabelled in-domain-data (More than 1.17 millions token) and 1 million token from WSJ. In addition, they used regular expressions to perfectly tag Twitter-specific tags: retweets, @usernames, hashtags, and urls, contrariwise our model which is end-to-end and do not use hand-crafted rules.

4.2. Transfer Learning Performances

In this section, we analyze the importance of each layer of our model in the transfer learning. Instead of transferring all layers, we experiment with transferring different combinations of layers. The objective is to understand which layers are the most transferable. It’s well known that the lowest layers of the DNN tend to represent domain-independent features, whereas the topmost layers are more domain-specific (Mou et al., 2016). So, we tried to transfer the features starting from the bottom-most layer up to the topmost layer (Except last softmax layer), adding one layer at a step.

Table 2 shows accuracy on T-POS data-set after transferring the features learned at each layer of the model trained on WSJ + IRC and also accuracy on ARK data-set after transferring the features learned at each layer of the model trained on T-train + WSJ + IRC (We fine tune layers features during the training).

Transferred layer	Acc. on T-POS (%)	Acc. on ARK (%)
Char-emb. (CNN)	89.66	90.96
Word-emb.	90.01	91.34
GRU	90.39	**91.66**
Fully-connected	**90.46**	91.00

Table 2: Effects of transferring different neural layers.

We can observe that transferring embedding and GRU layers improves significantly the accuracy on both data-sets despite the difference on the-tag set in the case of ARK data-set. However, transferring the fully connected layer improves the performance for T-POS data-set but degrades it for ARK data-set. A possible reason, is that in the fully connected layer, the model started to learn features that depend on the tag-set.

4.2.1. Pre-trained word level embedding performances

In order to test the importance of pretrained word embeddings, we perform experiments using different word embeddings (3.3.) to initialize words vectors, as well as a random initialization method. Table 3 shows our model’s accuracy on T-POS by initializing words embedddings with different pretrained words vectors as well as the random initialization. Two different settings are experimented. In the first ”Fine-tune”, weights are updated during training. In the second ”Freeze”, embedding weights are not updated. We can observe that the GloVe unsupervised vectors give the best score on both data-sets, and Word2vec gives the worse scores. One possible reason is that Word2Vec is not as good as the other embeddings because of vocabulary mismatch (as Word2Vec was trained on news data).

According to the results in table 3, models using pre-trained word embeddings obtain a significant improvement opposed to the ones using random embeddings. We also observe that fine tuning embedding vectors increases the performance, often significantly. In addition, we observe that combining the four embeddings vectors obtained with

[6] For ARK data-set, it is not possible to perform a joint-training with out-of-domain data because of the tag-set mismatch.

Initialization method	Freeze (%)	Fine-tune (%)
Random	–	76
Word2vec (1)	74	85
FastText (2)	82.2	85.7
Glove (3)	86	86.5
Glove-twitter (4)	82.6	85.5
Concatenation (1,2,3,4)	84.7	**89.02**

Table 3: Token accuracy of our model (trained jointly on T-train + WSJ + IRC) on T-POS data set with the initialization and the freezing or fine-tuning of word level embedding layer with different word embedding pre-trained models.

initialization from four pre-trained models gives the best performance of our model.

5. Conclusion

In this paper, we proposed a truly end-to-end Deep Neural Network (DNN) model for English Tweets POS tagging, with the problem of the unavailability of annotated data. We showed that word representations are crucially important for the success of DNN models. We also explored transfer learning to improve Twitter POS tagging by handling the problem of tweets annotated data availability and differences on tag-sets between different available corpora. This study offers several open issues for future work. First, we plan to perform more experiments on the transferability of different layers of DNN models. The second perspective is to study how the similarity between the source and target languages could influence the success of transfer learning.

6. Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 700381.

7. References

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Chiu, Jason PC and Eric Nichols, 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Choi, Jinho D, 2016. Dynamic feature induction: The last gist to the state-of-the-art. In *HLT-NAACL*.

Derczynski, Leon, Alan Ritter, Sam Clark, and Kalina Bontcheva, 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*.

Forsythand, Eric N and Craig H Martell, 2007. Lexical and discourse analysis of online chat dialog. In *Semantic Computing, 2007. ICSC 2007. International Conference on*. IEEE.

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith, 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics.

Gui, Tao, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang, 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu, 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Lee, Ji Young, Franck Dernoncourt, and Peter Szolovits, 2017. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*.

Ma, Xuezhe and Eduard Hovy, 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Marcus, Mitchell P, Mary Ann Marcinkiewicz, and Beatrice Santorini, 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Mou, Lili, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin, 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*.

Owoputi, Olutobi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith, 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Pan, Sinno Jialin and Qiang Yang, 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning, 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14.

Ritter, Alan, Sam Clark, Oren Etzioni, et al., 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer, 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics.

Yang, Zhilin, Ruslan Salakhutdinov, and William W Cohen, 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight, 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.