Alleviating Manual Feature Engineering for Part-of-Speech Tagging of Twitter Microposts using Distributed Word Representations

Fréderic Godin
Multimedia Lab
Ghent University - iMinds
Ghent, Belgium
frederic.godin@ugent.be

Baptist Vandersmissen
Multimedia Lab
Ghent University - iMinds
Ghent, Belgium
baptist.vandersmissen@ugent.be

Azarakhsh Jalalvand
Multimedia Lab
Ghent University - iMinds
Ghent, Belgium
azarakhsh.jalalvand@ugent.be

Wesley De Neve
Multimedia Lab & IVY Lab
Ghent University & KAIST
Ghent, Belgium & Daejeon, South Korea
wesley.deneve@ugent.be

Rik Van de Walle
Multimedia Lab
Ghent University - iMinds
Ghent, Belgium
rik.vandewalle@ugent.be

Abstract

Many algorithms for natural language processing rely on manual feature engineering. In this paper, we show that we can achieve state-of-the-art performance for part-of-speech tagging of Twitter microposts by solely relying on automatically inferred word embeddings as features and a neural network. By pre-training the neural network with large amounts of automatically labeled Twitter microposts to initialize the weights, we achieve a state-of-the-art accuracy of 88.9% when tagging Twitter microposts with Penn Treebank tags.

1 Introduction

Part-of-Speech (PoS) tagging is an important task in the field of Natural Language Processing (NLP), sitting at the core of algorithms such as sentiment analysis [1] and named entity recognition [2]. However, PoS tagging for microposts on online social networks still poses a significant challenge, given the short, unstructured and noisy nature of these microposts. While the Stanford PoS tagger [3] reaches an accuracy of 97.24% on news articles of the Wall Street Journal corpus, it only reaches an accuracy of 73.37% when tagging Twitter microposts [4]. Therefore, researchers [4, 5, 6, 7] came up with many carefully selected, micropost-specific features such as slang dictionaries, Brown clusters trained on Twitter data and phonetic normalization. These features allow boosting the accuracy of PoS tagging for Twitter microposts with Penn TreeBank (PTB) tags to 88.69% [4].

Despite this significant improvement in accuracy, these features are tailored for the task at hand, thus requiring manual interference when new types of slang emerge, or when another language or corpus is used. Therefore, we propose to make use of distributed word representations as features, elevating the task of manual feature engineering. These word embeddings can be used as an input to a feed-forward neural network [8]. To cope with the limited amount of training data, compared to the number of parameters in the neural network, the neural network can be pre-trained with large amounts of high-confidence annotated data, constructed via vote-constrained bootstrapping [4].

The main contribution of this paper is as follows: we demonstrate that we can achieve state-of-the-art PoS results without manual feature engineering by using large amounts of unlabeled microposts for constructing distributed word representations that capture syntactic and semantic regularities.

2 Related Work

Current state-of-the-art Part-of-Speech (PoS) taggers for news articles reach accuracies of up to 97.5% [9], using Penn TreeBank (PTB) tags [10]. Many PoS taggers [3, 9, 11, 12, 13] rely on similar features such as suffixes and affixes, bi-grams and tri-grams, and other lexical features such as uppercase characters. To achieve better results, some of them [9, 11] use semi-supervised methods where they use their current system to tag unlabeled data and then use this tagged data as input again to continue training their system . However, Collobert *et al.* [8] showed that they could build a PoS tagger from scratch that did not need handcrafted features to achieve state-of-the-art results. By training a neural network on the entire English Wikipedia, they inferred meaningful distributed word representations. Next, by concatenating a window of per-word feature vectors as input feature vector, they trained a neural network to predict the PoS tag of the central word of the window. They achieved an accuracy of 97.14% without manual feature engineering. Recently, Mikolov *et al.* [14] introduced a new type of word embedding that proved to be very powerful in capturing both syntactic and semantic relationships in English language, while also allowing for fast training.

Gimpel *et al.* [6] were among the first to design a PoS tagger for Twitter microposts. To do so, they annotated 1827 Twitter microposts with 25 tags that are more tailored for microposts than the PTB tags. They fed a Conditional Random Field (CRF) model with traditional features such as suffixes up to length three but also Twitter specific features such as phonetic normalization of the tokens and regular expressions for detecting hashtags and URLs. To deal with the lack of annotated data, they calculated distributional features for the 10,000 most common words. They finally reached an accuracy of 88.89% [7]. Owoputi *et al.* [7] improved upon this result by using a Maximum Entropy Markov Model (MEMM), instead of a CRF. They made use of Brown clusters, name lists and most-common-tag dictionaries. Their system yielded a 91.6% accuracy on the same test set.

Ritter *et al.* [5], on the other hand, proposed a PoS tagger that used the 48 PTB tags [10], extended with 4 additional tags for Twitter specific orthography. They made use of a CRF model, feeding it with traditional features such as contextual features and spelling features, as well as with PoS dictionaries and Brown clusters. They obtained an accuracy of 84.55% on an independent test set [4]. Derczynski *et al.* [4] improved this accuracy by adding new features as input to the Stanford tagger and providing more training data. Among others, they constructed a library that converts slang or erroneous words, and used dictionaries containing the most probable PoS tag per word. A significant boost in accuracy was realized by using substantial amounts of automatically annotated data, created by combining the output of multiple PoS taggers, i.e. their own tagger and the one of Owoputi *et al.* [7]. Their final method yielded an accuracy of 88.89% on the extended PTB tag set.

Similar to traditional PoS tagging for news articles, all of the aforementioned techniques for PoS tagging of Twitter microposts used manual feature engineering [4, 5, 6, 7]. A number of them [5, 6, 7] used word representations trained on large amounts of unlabeled Twitter microposts, namely distributional and cluster-based word representations [15]. In this paper, we *solely* make use of the recently introduced *distributed* word representations of Mikolov *et al.* [14]. Furthermore, based on the promising results of Collobert *et al.* [8], we make use of a neural network instead of a sequence model such as CRF or MEMM.

3 Proposed Method

Figure 1 gives a high-level overview of the proposed architecture, and where this architecture consists of a standard one-hidden-layer feed-forward neural network. The goal is to predict the PoS tag of a word $w(t)$ within a micropost. To do this, the words within a context window around $w(t)$ are used. This context window consists of an odd number of words, centered at $w(t)$. However, we do not feed these words directly to the neural network. Instead, we first transform the words into word embeddings that can be inferred in advance. To that end, we use the recently introduced word embeddings of Mikolov *et al.* [14], which are fast to train and which proved to be effective in capturing syntactic and semantic relationships in natural language. By using their word2vec software package, we can infer word embeddings w_{w2v_e} of millions of freely-available microposts for every word. Next, these per-word feature representations within the context window can be concatenated into a single feature vector that represents the neural network input vector for predicting the PoS tag of the central word $w(t)$.

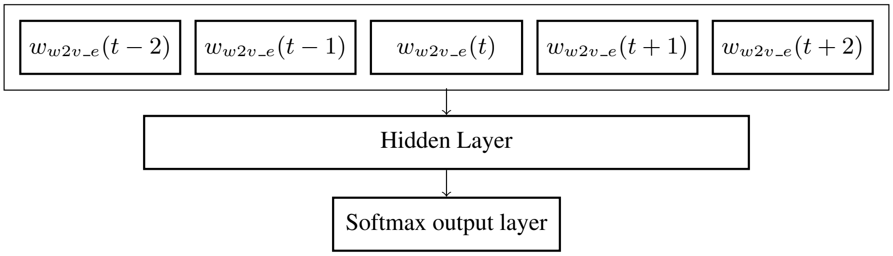

Figure 1: High-level architecture of the PoS tagger. Five word embeddings are concatenated and fed into a one-hidden-layer feed-forward neural network. The output consists of the PoS tag of $w(t)$.

Since only a limited amount of annotated microposts are available, we can use the technique of vote-constrained bootstrapping proposed by Derczynski *et al.* [4] to increase the amount of PoS labeled microposts. The idea of this technique is that high-confidence labeled data are created by only using microposts for which a number of PoS taggers agree. Given that those existing PoS taggers use different tag sets, Derczynski *et al.* defined a mapping between the tags of the different PoS taggers. Only if all PoS taggers agreed on the tags of the entire micropost, they kept the micropost. However, such a dataset only contains examples for which all taggers agree, which leads to a bias towards certain sequences of tokens or PoS tags. In other words, this additional training data was not representative for the original, manually annotated training data and contained errors. Therefore, we only use these automatically annotated microposts for pre-training the neural network in order to set the initial weights (instead of using random weights). Finally, we fine-tune the weights of the neural network using accurate, but scarcely-available manually annotated microposts.

4 Experiments

In order to test the proposed method, we focused on Part-of-Speech (Pos) tagging for Twitter microposts using the extended Penn TreeBank (PTB) tags of Ritter *et al.* [5]. We used their manually annotated dataset of 800 microposts (16K tokens). We used the 70:15:15 splits of Derczynski *et al.* [4] for training, validating and testing. Hence, our baselines on the test set are the initial algorithm of Ritter *et al.* [5] and the current state-of-the-art algorithm of Derczynski *et al.* [4], having accuracies of 84.55% and 88.69%, respectively [4].

Given that all current Twitter micropost PoS taggers used the same tokenizer, we also used this tokenizer[1] for tokenizing all Twitter microposts. Additionally, we used replacement tokens for URLs ($\_URL\_$), mentions ($\_MENTION\_$) and numbers ($\_NUMBER\_$). However, we did not replace hashtags as replacing hashtags experimentally showed to have a negative influence on the accuracy.

To train the PoS tagger, we first needed to construct the word embedding look-up table using word2vec[2]. To do so, we collected Twitter microposts from 300 days during the period 1/03/2013 - 28/02/2014. We filtered out non-English microposts by using the micropost language classifier of Godin *et al.* [16]. Then, we applied the tokenization step as described in the previous paragraph and fed over 400 million tokenized microposts to the word2vec software to infer word embeddings.

Next, we trained the neural network with the training set of Ritter *et al.* while varying the size of the word2vec word embeddings, the context window and the number of hidden nodes in the neural network. We used mini-batch learning with a batch size of 20, applied L2-regularization on the weights and used a *tanh* activation function for the hidden units. We iterated and updated the weights using stochastic gradient descent until the validation accuracy did not improve anymore.

The best results obtained on the validation and test set are depicted in Table 1. The skip-gram architecture in combination with the negative sampling approach and a window of three [14] consistently gave the best word embeddings. The best result on the independent test set without pre-training the neural network was an accuracy of 86.9%. This accuracy is already higher than the initial implementation of Ritter *et al.* (84.55%) but still lower than Derczynski *et al.* (88.69%).

[1] Python port of Brendan O'Connor's tokenizer: https://github.com/myleott/ark-twokenize-py

[2] Mikolov's word2vec: https://code.google.com/p/word2vec/

The limited amount of high-quality data impeded the further improvement of the result significantly. In fact, the training set only contains 10K examples while the proposed neural network typically has 100K to 1M parameters. Therefore, to pre-train the network, we made use of large amounts of automatically labeled microposts using the technique of vote-constrained bootstrapping [4]. We used the dataset provided by Derczynski *et al.*, which contains 160K automatically annotated microposts having on average 9.7 tokens per micropost. We pre-trained the network with the first 50K and 125K microposts[3] and used the last 1K microposts as our validation set. Next, to initialize the neural network, we used the weights for which the best result was obtained on the 1K validation set, and trained the neural network on the training set of Ritter *et al.* using the same procedure as described in the previous paragraph.

As depicted in Table 1, pre-training the network with 50K automatically labeled microposts improved the result for both our validation and test set with 1.5%, from 88.22% to 89.73% and from 86.9% to 88.46%, respectively. When we enlarged the training dataset for the word2vec algorithm from 150 million microposts to 400 million microposts, the same result was obtained on the validation set but interestingly enough, the accuracy on the test set improved. The final step was to enlarge the pre-training dataset from 50K microposts to 125K microposts. This extra pre-training data boosted the accuracy on the validation set over 90%. Hence, the final accuracy on the test set was 88.9%, which beats the current state-of-the-art accuracy of Derczynski *et al.* of 88.69%.

Table 1: Results obtained for the proposed architecture with and without pre-training.

Word2vec		**Neural network**				
Dataset size (microposts)	**Vector size**	**Context window**	**Input size**	**Hidden units**	**Validation accuracy**	**Test accuracy**
No pre-training						
150M	400	3	1200	400	**88.22%**	**86.90%**
150M	400	3	1200	500	87.95%	87.46%
Pre-trained with 50K microposts						
150M	400	3	1200	500	89.64%	88.82%
150M	200	5	1000	400	**89.73%**	**88.46%**
400M	400	3	1200	500	**89.73%**	**88.95%**
400M	200	5	1000	400	89.60%	88.42%
Pre-trained with 125K microposts						
400M	400	3	1200	500	**90.09%**	**88.90%**
400M	400	3	1200	600	90.00%	89.08%

5 Conclusion and Outlook

In this paper, we showed that we can achieve state-of-the-art performance for Part-of-Speech tagging of microposts by only using word embeddings as features and a neural network for predicting a corresponding PoS tag. By using a data-driven approach for inferring good features and training the neural network, we were able to boost the accuracy up to 88.9%. This is slightly higher than the current state-of-the-art algorithm [4] that uses many hand-engineered features available in the Stanford Tagger and several dictionaries to deal with spelling mistakes and slang. Our proposed method, on the other hand, uses automatically inferred word embeddings that capture syntactic and semantic patterns, not needing manual feature engineering. Moreover, the proposed technique can be easily applied to many different (Latin) languages or within different domains without having to revise the features used; only the word embeddings need to be retrained on a large corpus of data.

In the future, we would like to investigate other techniques for pre-training neural networks and for inferring word embeddings. We would like to investigate techniques such as stacked denoising auto-encoders for pre-training multiple layers, layer-by-layer. Preliminary experiments with multiple layers and large amounts of automatically labeled data did not (yet) yield satisfactory results. We also would like to investigate the influence of the word representations used, given that the best word representation is task dependent [15]. Therefore, it would be interesting to investigate the application of other word representations for PoS tagging of microposts.

[3] Due to the limited memory within the GPU used, it was impossible to load all 160K microposts

Acknowledgment

The research activities described in this paper were funded by Ghent University, iMinds, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

[1] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242, 2013.

[2] Amparo Elizabeth Cano Basave, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. Making sense of microposts (#msm2013) concept extraction challenge. In *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, pages 1–15, 2013.

[3] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *proceedings of HLT-NAACL*, pages 252–259, 2003.

[4] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206. RANLP 2011 Organising Committee / ACL, 2013.

[5] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[6] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 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*, HLT '11, pages 42–47, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[7] Olutobi Owoputi, Brendan OConnor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390, 2013.

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011.

[9] Anders Søgaard. Semisupervised condensed nearest neighbor for part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 48–52, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

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

[11] Drahomíra "johanka" Spoustová, Jan Hajič, Jan Raab, and Miroslav Spousta. Semi-supervised training for the averaged perceptron pos tagger. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 763–771, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[12] Libin Shen, Giorgio Satta, and Aravind Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[13] Jess Gimnez and Llus Mrquez. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 43–46, 2004.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[15] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[16] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 593–596, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.