NER for Medical Entities in Twitter using Sequence to Sequence Neural Networks

Antonio Jimeno Yepes[◇] **and Andrew MacKinlay**[◇♠]
[◇] IBM Research – Australia, Melbourne, VIC, Australia
[♠] Dept of Computing and Information Systems, University of Melbourne, Australia
{antonio.jimeno,admackin}@au1.ibm.com

Abstract

Social media sites such as Twitter are attractive sources of information due to their combination of accessibility, timeliness and large data volumes. Identification of medical entities in Twitter can support tasks such public health surveillance. We propose an approach to perform annotation of medical entities using a sequence to sequence neural network. Results show that our approach improves over previous work based on CRF in the annotation of two medical entities types in Twitter.

1 Introduction

Public health surveillance (Nsubuga et al., 2006) is the systematic collection, analysis and monitoring of population health for the public good using a variety of tools. Governments rely mostly on aggregated health data from health care centres, which cannot be used in real-time scenarios such as syndromic surveillance.

Twitter is a promising source of information due to its availability and the large quantity of information published, with more than 500 million posts published each day. It can be potentially used to mine information about the status of the population (Jimeno Yepes et al., 2015a), and can then be used in combination with other sources to empower government decision makers.

Natural language processing can leverage the short messages in Twitter by extracting pieces of structured information from them which can be aggregated for data analysis. Identifying entities of interest in tweets is relevant for several tasks, including public health surveillance. Several approaches have been used to perform named entity recognition in Twitter, which range from dictionary matching approaches to machine learning methods. Named entity recognition (NER) in Twitter has unique challenges compared to many other sources of text. Tweets are short (with at most 140 characters), and often contain highly informal language, idioms, humour, typos and grammatical errors (Baldwin et al., 2013; Jimeno Yepes et al., 2015b).

Effective biomedical NER methods for Twitter rely on machine learning methods such as conditional random fields (CRF) (Lafferty et al., 2001). Learning algorithms like CRF typically consider a limited span of one token before and/or after the token being annotated, which may omit valuable contextual information in order to work within the limitations of the learning algorithms.

We use a recurrent neural network under the assumption that it can manage dependencies in language better than traditional methods such as CRF. The proposed approach uses a sequence to sequence network (Sutskever et al., 2014) to analyse text in a tweet producing an encoding vector which is then used to perform the annotation by a second LSTM (Long Short Term Memory) node. Word embeddings generated from several million tweets are used to represent the tokens in text, which is the only feature engineering required in this neural network approach.

This approach improves over previous work based on CRF in the annotation of two medical entity types in Twitter. It also takes advantage of word embeddings, thus reducing the domain-specificity of the NE recogniser, i.e. no domain-specific lexicon is used by the LSTM approach.

2 Methods

In this section, we describe the generation of word embeddings and the recurrent network structure used. We describe the data set used and the CRF baseline method.

Antonio Jimeno Yepes and Andrew MacKinlay. 2016. NER for Medical Entities in Twitter using Sequence to Sequence Neural Networks. In *Proceedings of Australasian Language Technology Association Workshop*, pages 149−153.

2.1 Word embeddings

Bag-of-words representations for natural language processing are typically high-dimensional (with dimension equal to the vocabulary size) and very sparse, since just the words in the represented document will have non-zero values. This representation is a problem for deep neural networks and recently word embeddings have been used. Word embeddings map this high dimensional space into a lower dimensionality space which is dense rather than sparse and has some interesting properties that allow, for instance, identifying words with a similar meaning (Mikolov et al., 2013).

The data set used in this work to generate the word embeddings consists of 148 million tweets randomly selected from years 2012 and 2013. We used word2vec[1] to generate word vectors using the continuous bag-of-words (CBOW) approach (Mikolov et al., 2013) with 200 dimensions and other parameters set to default values.

Tweet text was lowercased and then tokenised using the TweetNLP package[2].

2.2 Sequence to sequence neural network

Typically recurrent neural networks suffer from the *vanishing gradient* problem (Bengio et al., 1994; Pascanu et al., 2013) when trained on long dependencies. These networks rely on gradient descent methods and *backpropagation through time*, thus gradients after long dependencies might be very small. As an effect, the time needed to train a network might be quite large. It can even make a problem difficult to learn since the signal for important events might be missed. LSTM (Hochreiter and Schmidhuber, 1997) memory cells were developed to overcome the vanishing gradient limitation and it can learn to retain information for a long period of time.

LSTM memory cells introduce mechanisms to avoid the vanishing gradient problem using, for a given time t, an input gate i_t, an output gate o_t, a forget gate f_t and a cell c_t. The weights for these three gates and memory cell are trained using backpropagation using training data. The input to the LSTM cell is the vector x_t and the hidden output is h_t. The ability of LSTM to effectively deal with long-range dependencies (such as syntactic dependencies) may be useful for NLP tasks such as disambiguation.

[1]https://code.google.com/archive/p/word2vec

[2]www.cs.cmu.edu/ ark/TweetNLP

We adopt the definition of LSTM memory cell introduced in (Graves, 2013), which follows the diagram in Figure 1. Equations 1 to 5 show how the values in different LSTM components get calculated. Weights matrices W have subscripts that indicate the components being related. For instance W_{hi} is the weight matrix between the hidden output and the input gate.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$
$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$
$$h_t = o_t tanh(c_t) \quad (5)$$

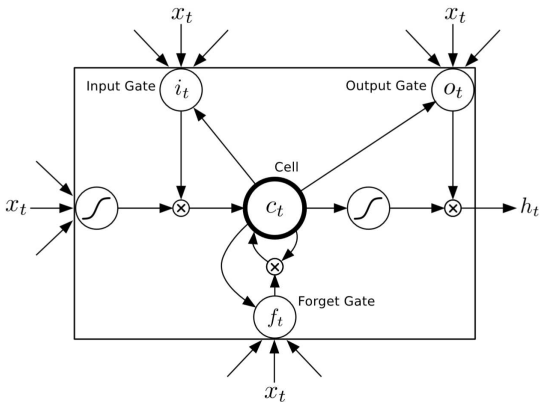

Figure 1: LSTM memory cell unit diagram (Graves, 2013)

Our network design is inspired by the previously-mentioned sequence to sequence learning network for machine translation of Sutskever et al. (2014).

In our network, we have two LSTM nodes with n units each. The first one (LSTMe) is used encoded to encode the text into a vector representation. The second one (LSTMa) takes as input an input vector and the current token and generates an annotation. The output of LSTMa is processed by a linear classifier trained using multiclass hinge loss. In this stage, one of three categories is predicted, corresponding to IOB tags common in NER: B(eginning), I(nside entity) and O(ut of entity).

Training relies on AdaGrad (Duchi et al., 2011), and the learning rate has been set to 0.01 for all iterations, with the number of iterations set to 300.

A lookup table was used to translate each token into a vector representation. Word embedding vec-

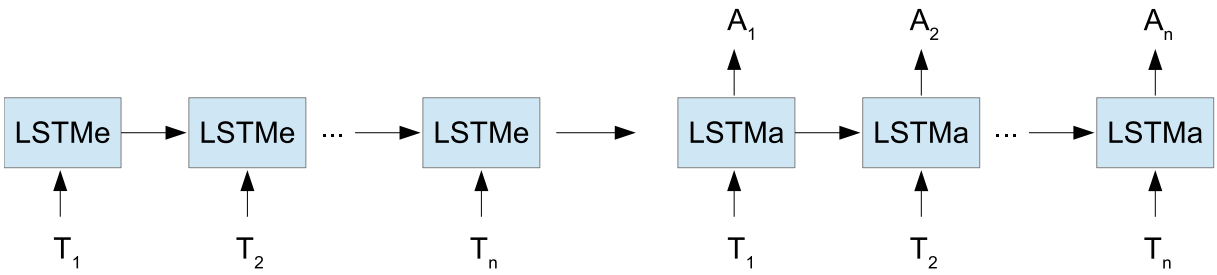

Figure 2: Recurrent network layout. LSTMe performs encoding of tweets while LSTMa performs B(eginning), I(n entity) and O(ut of entity) annotation.

tors of 200 dimensions for each token were generated. The LSTM memory cells have been configured to use 200 units each, which matches the dimensions of the vectors generated with word2vec.

The vocabulary for word embeddings is finite, as it is dependent on the training data we supply to word2vec training data; in addition, words with less than 20 occurrences in the training set are removed to reduce the size of the vocabulary. This means that some tokens in the NER training and test sets might not be found in the word embedding lookup table. In these cases, a vector with all dimensions equal to zero is provided to the LSTM nodes for both training and evaluation.

2.3 Micromed data set

We have used the Micromed[3] data set described in Jimeno Yepes et al. (2015b), which contains 1300 tweets from May 2014 manually annotated with mentions of diseases, pharmacological substances and symptoms. In addition, the annotations include parts-of-speech for some of the entities, since adjectives have been considered for symptoms and it includes references to either figurative terms or entities that look as medical terms but are used in a figurative meaning (e.g. mentions of *heart attack* when someone is anxious).

Non medical entities were removed and the JSON format was converted into IOB format for training the recurrent network system based on Torch[4] and then converted into BRAT format[5] for evaluation. As in the generation of word embeddings, Tweet text was lowercased and then tokenized using the TweetNLP package.

[3]https://github.com/IBMMRL/medinfo2015
[4]http://torch.ch
[5]http://brat.nlplab.org/standoff.html

2.4 CRF baseline method

As an informed comparative baseline, we used previously published work (Jimeno Yepes et al., 2015a) based on a linear-chain CRF system with the following features: part-of-speech, token surface form and window relative position, token prefix and suffix character *n*-grams of all lengths up to eight, whether the token appears in a list of terms extracted from the UMLS (Unified Medical Language System) (Bodenreider, 2004) corresponding to the entity type. This is currently the method with best performance on this data set.

3 Results

The Micromed data set is split in 13 subsets of 100 tweets each, which are used to train and test the machine learning methods using 13-fold cross validation.

Results are presented in tables 1 and 2. Previous results published on Micromed (Jimeno Yepes et al., 2015a) are compared to the proposed LSTM approach with word embeddings as features. We report precision, recall and F1 results for each method, for both exact match (boundaries exactly match the gold standard) and partial match (the postulated entity overlaps at all with the gold standard). Statistical significance was determined using a randomisation version of the paired sample *t*-test (Cohen, 1996).

Our approach improves the performance for both pharmacological substance and symptom entity types against the CRF comparison method. However, the performance on the disease entity type is decreased, particularly with exact match.

4 Discussion

The proposed method improves the F-score in the annotation of pharmacological substances due to a boost in recall. It improves the accuracy on the

	Disease			Pharm.Substance			Symptom		
Method	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Baseline	**0.796**	**0.527**	**0.634**†	**0.821**	0.396	0.535	0.720	0.608	0.659
LSTMe+LSTMa	0.600	0.439	0.507	0.694	**0.476**	**0.565**	**0.744**	**0.619**	**0.676**

Table 1: Exact match NER results. Precision (Prec), recall (Rec) and F1 are used for evaluation. Baseline tagger and the proposed approach (LSTMe+LSTMa) are trained and evaluated for the three entity types: disease, pharmacological substance (Pharm.Substance) and symptom. † denotes statistically significant different of $p < 0.05$ against the alternate method.

	Disease			Pharm.Substance			Symptom		
Method	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Baseline	**0.845**	0.562	**0.675**	**0.854**	0.415	0.558	0.746	0.630	0.683
LSTMe+LSTMa	0.795	**0.572**	0.665	0.838	**0.575**	**0.682**†	**0.793**	**0.656**	**0.718**†

Table 2: Partial match Named entity recognition results. Precision (Prec), recall (Rec) and F1 are used for evaluation. Baseline tagger and the proposed approach (LSTMe+LSTMa) are trained and evaluated for the three entity types: disease, pharmacological substance (Pharm.Substance) and symptom. † denotes statistically significant difference of $p < 0.05$ against the alternate method.

annotation of symptoms to a lesser extent. On the other hand, the accuracy decreases for the annotation of diseases, mostly in the exact match evaluation.

A manual examination of the neural network annotation shows that for entities with multiple tokens, annotations are sometimes incorrect for one of the tokens in the entity.

The recall is typically higher in our NER system, even though the CRF baseline uses external domain-specific terminological resources such as the UMLS. In fact, the CRF baseline heavily relies on the term lists extracted from the UMLS, which represent extensive domain-specific customisation, while our LSTM approach does not depend on anything specific to the medical domain apart from the relatively small training corpus. The external knowledge comes solely from word embeddings derived from a general Twitter corpus, which would only contain a very small amount of domain-specific information, suggesting that the LSTM approach is more robust and domain-agnostic.

Even with higher recall, there are some terms that are missed by the neural network tagger and a larger data set or domain knowledge based on the UMLS could provide a boost in the recall. Infrequent terms might cause problems – for example, *valerian* did not appear in the word embeddings generated using word2vec. A larger Twitter corpus and/or a domain corpus based on MEDLINE might be considered to generate the word embeddings to make them more robust and encode more relevant domain knowledge including rarer phenomena such as these.

5 Related work

CRF has become the standard tool for NER, including for Twitter (Ritter et al., 2011; Ritter et al., 2012). Previous work in biomedical natural language processing (NLP) has explored using CRF methods to several manually annotated data sets (Jimeno Yepes et al., 2015b; Nikfarjam et al., 2015).

Initial work using convolutional neural networks (Collobert et al., 2011) showed state-of-the-art performance on standard general English data sets. There is recent work on named entity recognition using bidirectional LSTM based recurrent networks that are enhanced using CRFs (Lample et al., 2016) and complemented with CNNs (Ma and Hovy, 2016), which show state-of-the-art performance in standard data sets. These works use word-based features as we do here but also features at the character level.

6 Conclusions and Future work

The proposed method improves the annotation of biomedical entities in Twitter based on a previously state-of-the-art method based on CRF. For future work, we may consider additional neural network designs. For example, to alleviate the aforementioned problem of tokens only partially

aligning with ground truth, we could use methods discussed in a preprint (Lample et al., 2016) to combine the output of a bigraph neural network with CRF to improve the annotation of the beginning of the entity (B). We may also consider additional neural network designs inspired by other work on deep learning for NER, and combining external domain knowledge from a terminology such as the UMLS, which may improve accuracy while making it more targeted to the medical domain and this particular corpus.

NER over medical entities is a potentially useful part of a public health surveillance system; further normalisation of these terms, and detecting figurative or non-medical uses would further enhance their utility and could also benefit from a deep neural network approach.

References

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *IJCNLP*, pages 356–364.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

Paul R Cohen. 1996. Empirical methods for artificial intelligence. *IEEE Intelligent Systems*, (6):88.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

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

Antonio Jimeno Yepes, Andrew MacKinlay, and Bo Han. 2015a. Investigating public health surveillance using twitter. *ACL-IJCNLP 2015*, page 164.

Antonio Jimeno Yepes, Andrew MacKinlay, Bo Han, and Qiang Chen. 2015b. Identifying diseases, drugs, and symptoms in twitter. In *MEDINFO 2015: eHealth-enabled Health - Proceedings of the 15th World Congress on Health and Biomedical Informatics, São Paulo, Brazil, 19-23 August 2015*, pages 643–647.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

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

Tomas Mikolov, 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*, pages 3111–3119.

Azadeh Nikfarjam, Abeed Sarker, Karen OConnor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681.

Peter Nsubuga, Mark E. White, Stephen B. Thacker, Mark A. Anderson, Stephen B. Blount, Claire V. Broome, Tom M. Chiller, Victoria Espitia, Rubina Imtiaz, Dan Sosin, Donna F. Stroup, Robert V. Tauxe, Maya Vijayaraghavan, and Murray Trostle. 2006. Public health surveillance: a tool for targeting and monitoring interventions. In *Disease Control Priorities in Developing Countries. 2nd edition.* World Bank.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *EMNLP*.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *KDD*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.