Passive-Aggressive Sequence Labeling with Discriminative Post-Editing for Recognising Person Entities in Tweets

Leon Derczynski
University of Sheffield
leon@dcs.shef.ac.uk

Kalina Bontcheva
University of Sheffield
kalina@dcs.shef.ac.uk

Abstract

Recognising entities in social media text is difficult. NER on newswire text is conventionally cast as a sequence labeling problem. This makes implicit assumptions regarding its textual structure. Social media text is rich in disfluency and often has poor or noisy structure, and intuitively does not always satisfy these assumptions. We explore noise-tolerant methods for sequence labeling and apply discriminative post-editing to exceed state-of-the-art performance for person recognition in tweets, reaching an F1 of 84%.

1 Introduction

The language of social media text is unusual and irregular (Baldwin et al., 2013), with misspellings, non-standard capitalisation and jargon, disfluency and fragmentation. Twitter is one of the sources of social media text most challenging for NLP (Eisenstein, 2013; Derczynski et al., 2013).

In particular, traditional approaches to Named Entity Recognition (NER) perform poorly on tweets, especially on person mentions – for example, the default model of a leading system reaches an F1 of less than 0.5 on person entities in a major tweet corpus. This indicates a need for approaches that can cope with the linguistic phenomena apparently common among social media authors, and operate outside of newswire with its comparatively low linguistic diversity.

So, how can we adapt? This paper contributes two techniques. Firstly, it demonstrates that entity recognition using noise-resistant sequence labeling outperforms state-of-the-art Twitter NER, although we find that recall is consistently lower than precision. Secondly, to remedy this, we introduce a method for automatically post-editing the resulting entity annotations by using a discriminative classifier. This improves recall and precision.

2 Background

Named entity recognition is a well-studied problem, especially on newswire and other long-document genres (Nadeau and Sekine, 2007; Ratinov and Roth, 2009). However, experiments show that state-of-the-art NER systems from these genres do not transfer well to social media text.

For example, one of the best performing general-purpose named entity recognisers (hereon referred to as Stanford NER) is based on linear-chain conditional random fields (CRF) (Finkel et al., 2005). The model is trained on newswire data and has a number of optimisations, including distributional similarity measures and sampling for remote dependencies. While excellent on newswire (overall F1 90%), it performs poorly on tweets (overall F1 44%) (Ritter et al., 2011).

Rule-based named entity recognition has performed a little better on tweets. Another general-purpose NER system, ANNIE (Cunningham et al., 2002), reached F1 of 60% over the same data (Derczynski et al., 2013); still a large difference.

These difficulties spurred Twitter-specific NER research, much of which has fallen into two broad classes: semi-supervised CRF, and LDA-based.

Semi-supervised CRF: Liu et al. (2011) compare the performance of a person name dictionary (F1 of 33%) to a CRF-based semi-supervised approach (F1 of 76% on person names), using a dataset of 12 245 tweets. This, however, is based on a proprietary corpus, and cannot be compared to, since the system is also not available.

Another similar approach is TwiNER (Li et al., 2012), which is focused on a single topic stream as opposed to general-purpose NER. This leads to high performance for a topic-sensitive classifier trained to a particular stream. In contrast we present a general-purpose approach. Further, we extract a specific entity class, where TwiNER performs entity chunking and no classification.

LDA and vocabularies: Ritter et al. (2011)'s T-NER system uses 2,400 labelled tweets, unlabelled data and Linked Data vocabularies (Freebase), as well as co-training. These techniques helped but did not bring person recognition accuracy above the supervised MaxEnt baseline in their experiments. We use this system as our baseline.

3 Experimental Setup

3.1 Corpus

The experiments combine person annotations from three openly-available datasets: Ritter et al. (2011), UMBC (Finin et al., 2010) and MSM2013 (Basave et al., 2013). In line with previous research (Ritter et al., 2011), annotations on @mentions are filtered out. The placeholder tokens in MSM data (i.e. _MENTION_, _HASHTAG_, _URL_) are replaced with @Mention, #hashtag, and http://url/, respectively, to give case and character n-grams more similar to the original values.

The total corpus has 4 285 tweets, around a third the size of that in Liu et al. (2011). This dataset contains 86 352 tokens with 1 741 entity mentions.

Person entity recognition was chosen as it is a challenging entity type. Names of persons popular on Twitter change more frequently than e.g. locations. Person names also tend to have a long tail, not being confined to just public figures. Lastly, although all three corpora cover different entity types, they all have Person annotations.

3.2 Labeling Scheme

Following Li et al. (2009) we used two-class IO labeling, where each token is either in-entity or out-of-entity. In their NER work, this performed better than the alternative BIO format, since data sparsity is reduced. The IO scheme has the disadvantage of being unable to distinguish cases where multiple different entities of the same type follow each other without intervening tokens. This situation is uncommon and does not arise in our dataset.

3.3 Features

The Stanford NER tool was used for feature generation. When required, nominal values were converted to sparse one-hot vectors. Features for modelling context are included (e.g. ngrams, adjoining labels). Our feature sets were:

base: default Stanford NER features, plus the previous and next token and its word shape.[1]

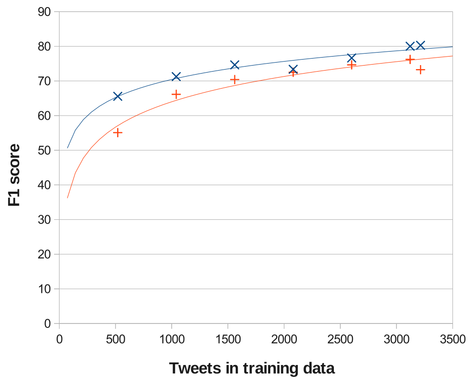

Figure 1: Training curve for *lem*. Diagonal cross (blue) is CRF/PA, vertical cross (red) SVM/UM.

lem: with added lemmas, lower-case versions of tokens, word shape, and neighbouring lemmas (in attempt to reduce feature sparsity & cope better with lexical and orthographic noise). Word shape describes the capitalisation and the type of characters (e.g. letters, numbers, symbols) of a word, without specifying actual character choices. For example, *Capital* may become *Ww*.

These representations are chosen to compare those that work well for newswire to those with scope for tolerance of noise, prevalent in Twitter.

3.4 Classifiers

For structured sequence labeling, we experiment with conditional random fields – CRF (Lafferty et al., 2001) – using the CRFsuite implementation (Okazaki, 2007) and LBFGS. We also use an implementation of the passive-aggressive CRF from CRFsuite, choosing *max_iterations = 500*.

Passive-aggressive learning (Crammer et al., 2006) demonstrates tolerance to noise in training data, and can be readily adapted to provide structured output, e.g. when used in combination with CRF. Briefly, it skips updates (is *passive*) when the hinge loss of a new weight vector during update is zero, but when it is positive, it aggressively adjusts the weight vector regardless of the required step size. This is integrated into CRF using a damped loss function and passive-aggressive (PA) decisions to choose when to update. We explore the PA-I variant, where the objective function scales linearly with the slack variable.

[1] Default plus useClassFeature=true, noMidNGrams=true, maxNGramLeng=6, usePrev=true, useNext=true, usePrevSequences=true, maxLeft=1, useTypeSeqs=true, useTypeSeqs2=true, useTypeSeqs3=true, useTypeySequences=true, wordShape=chris2useLC, useDisjunctive=true, lowercaseNGrams=true, useShapeConjunctions=true

Approach	Precision	Recall	F1
Stanford	85.88	50.00	63.20
Ritter	77.23	**80.18**	**78.68**
MaxEnt	**86.92**	59.09	70.35
SVM	77.55	59.16	67.11
SVM/UM	73.26	*69.63*	71.41
CRF	82.94	62.39	71.21
CRF/PA	80.37	65.57	*72.22*

Table 1: With base features (base)

Approach	Precision	Recall	F1
Stanford	90.60	60.00	72.19
Ritter	77.23	**80.18**	78.68
MaxEnt	**91.10**	66.33	76.76
SVM	88.22	66.58	75.89
SVM/UM	81.16	*74.97*	77.94
CRF	89.52	70.52	78.89
CRF/PA	86.85	74.71	**80.32**

Table 2: With shape and lemma features (lem)

For independent discriminative classification, we use SVM, SVM/UM and a maximum entropy classifier (MegaM (Daumé III, 2004)). SVM is provided by the SVMlight (Joachims, 1999) implementation. SVM/UM is an uneven margins SVM model, designed to deal better with imbalanced training data (Li et al., 2009).

3.5 Baselines

The first baseline is the Stanford NER CRF algorithm, the second Ritter's NER algorithm. We adapted the latter to use space tokenisation, to preserve alignment when comparing algorithms. Baselines are trained and evaluated on our dataset.

3.6 Evaluation

Candidate entity labelings are compared using the CoNLL NER evaluation tool (Sang and Meulder, 2003), using precision, recall and F1. Following Ritter, we use 25%/75% splits made at tweet, and not token, level.

4 Results

The base feature set performs relatively poorly on all classifiers, with only MaxEnt beating a baseline on any score (Table 1). However, all achieve a higher F1 score than the default Stanford NER. Of these classifiers, SVM/UM achieved the best precision and CRF/PA – the best F1. This demonstrates that the noise-tolerance adaptations to SVM and CRF (uneven margins and passive-aggressive updates, respectively) did provide improvements over the original algorithms.

Results using the extended features (lem) are shown in Table 2. All classifiers improved, including the baseline Stanford NER system. The SVM/UM and CRF/PA adaptations continued to outperform the vanilla models. With these features, MaxEnt achieved highest precision and CRF variants beat both baselines, with a top F1 of 80.32%. We continue using the *lem* feature set.

Entity length (tokens)	Count
1	610
2	1065
3	51
4	15

Table 3: Distribution of person entity lengths.

5 Discriminative Post-Editing

Precision is higher than recall for most systems, especially the best CRF/PA (Table 2). To improve recall, potential entities are re-examined in *post-editing* (Gadde et al., 2011). Manual post-editing improves machine translation output (Green et al., 2013); we train an automatic editor.

We adopt a gazetteer-based approach to triggering a discriminative editor, which makes decisions about labels after primary classification. The gazetteer consists of the top 200 most common names in English speaking countries. The first names of popular figures over the past two years (e.g. *Helle*, *Barack*, *Scarlett*) are also included. This gives 470 case-sensitive *trigger* terms.

Often the trigger term is just the first in a sequence of tokens that make up the person name. As can be seen from the entity length statistics shown in Table 3, examining up to two tokens covers most (96%) person names in our corpus. Based on this observation, we look ahead just one extra token beyond the trigger term. This gives a token sub-sequence that was marked as out-of-entity by the original NER classifier. Its constituents become *candidate* person name tokens.

Candidates are then labeled using a high-recall classifier. The classifier should be instance-based, since we are not labeling whole sequences. We chose SVM with variable cost (Morik et al., 1999), which can be adjusted to prefer high recall.

To train this classifier, we extract a subset of instances from the current training split as follows. Each trigger term is included. Also, if the trigger term is labeled as an entity, each subsequent in-entity token is also included. Finally, the next out-of-entity token is also included, to give examples of when to stop. For example, these tokens are either in or out of the training set:

		Overall		
Method	**Missed entity F1**	**P**	**R**	**F1**
No editing - plain CRF/PA	0.00	86.85	74.71	80.32
Naïve: trigger token only	5.82	86.61	78.91	82.58
Naïve: trigger plus one	6.05	81.26	82.08	81.67
SVM editor, $Cost = 0.1$	78.26	87.38	79.16	83.07
SVM editor, $Cost = 0.5$	89.72	87.17	80.30	83.60
SVM editor, $Cost = 1.0$	90.74	87.19	80.43	83.67
SVM editor, $Cost = 1.5$	92.73	87.23	80.69	**83.83**
SVM editor, $Cost = 2.0$	92.73	87.23	80.69	**83.83**

Table 4: Post-editing performance. Higher $Cost$ sacrifices precision for recall.

```
Miley    O       in
Heights  O       out

Miley    PERSON  in
Cyrus    PERSON  in
is       O       in
famous   O       out
```

When post-editing, the window is any trigger term and the following token, regardless of initial label. The features used were exactly the same as with the earlier experiment, using the *lem* set. This is compared with two naïve baselines: always annotating trigger terms as Person, and always annotating trigger terms and the next token as Person.

Results are shown in Table 4. Naïve editing baselines had F1 on missed entities of around 6%, showing that post-editing needs to be intelligent.

At $Cost = 1.5$, recall increased to 80.69, exceeding the Ritter recall of 80.18 (raising $Cost$ beyond 1.5 had no effect). This setup gave good accuracy on previously-missed entities (second column) and improved overall F1 to 83.83. It also gave better precision and recall than the best naïve baseline (trigger-only), and 6% absolute higher precision than trigger plus one. This is a 24.2% reduction in error over the Ritter baseline (F1 78.68), and a 17.84% error reduction compared to the best non-edited system (CRF/PA+*lem*).

6 Error Analysis

We examine two types of classification error: false positives (spurious) and false negatives (missed).

False positives occur most often where non-person entities are mentioned. This occurred with mentions of organisations (*Huff Post*), locations (*Galveston*) and products (*Exodus Porter*). Descriptive titles were also sometimes mis-included in person names (*Millionaire Rob Ford*). Names of persons used in other forms also presented as false positives (e.g. *Marie Claire* – a magazine). Polysemous names (i.e. words that could have other functions, such as a verb) were also mis-resolved (*Mark*). Finally, proper nouns referring to groups were sometimes mis-included (*Haitians*).

Despite these errors, precision almost always remained higher than recall over tweets. We use in-domain training data, and so it is unlikely that this is due to the wrong kinds of person being covered in the training data – as can sometimes be the case when applying tools trained on newswire.

False negatives often occurred around incorrect capitalisation and spelling, with unusual names, with ambiguous tokens and in low-context settings. Both omitted and added capitalisation gave false negatives (*charlie gibson*, or *KANYE WEST*). Spelling errors also led to missed names (*Russel Crowe*). Ambiguous names caused false negatives and false positives; our approach missed *mark* used as a name, and the surname of *Jack Straw*. Unusual names with words typically used for other purposes were also not always correctly recognised (e.g. *the Duck Lady*, or the last two tokens of *Spicy Pickle Jr.*). Finally, names with few or no context words were often missed (*Video: Adele 21.*, and *17-9-2010 Tal al-Mallohi, a 19-*).

7 Conclusion

Finding named entities in social media text, particularly tweets, is harder than in newswire. This paper demonstrated that adapted to handle noisy input is useful in this scenario. We achieved the good results using CRF with passive-aggressive updates. We used representations rich in word shape and contextual features and achieved high precision with moderate recall (65.57–74.71).

To improve recall, we added a post-editing stage which finds candidate person names based on trigger terms and re-labels them using a cost-adjusted SVM. This flexible and re-usable approach lead to a final reduction in error rate of 24.2%, giving performance well above that of comparable systems.

Acknowledgment This work received funding from EU FP7 under grant agreement No. 611233, Pheme. We thank Chris Manning and John Bauer of Stanford University for help with the NER tool.

References

T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. 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. ACL.

A. E. C. Basave, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. 2013. Making Sense of Microposts (# MSM2013) Concept Extraction Challenge. In *Proceedings of the Concept Extraction Challenge at the Workshop on 'Making Sense of Microposts'*, volume 1019. CEUR-WS.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: an Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175.

H. Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name#daume04cg-bfgs`, implementation available at `http://hal3.name/megam/`, August.

L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. 2013. Microblog-Genre Noise and Impact on Semantic Annotation Accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM.

J. Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369. Association for Computational Linguistics.

T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

P. Gadde, L. Subramaniam, and T. A. Faruquie. 2011. Adapting a WSJ trained part-of-speech tagger to noisy text: preliminary results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*. ACM.

S. Green, J. Heer, and C. D. Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 439–448. ACM.

T. Joachims. 1999. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight.joachims. org/, University of Dortmund*, 19(4).

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco: Morgan Kaufmann.

Y. Li, K. Bontcheva, and H. Cunningham. 2009. Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. *Natural Language Engineering*, 15(2):241–271.

C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. 2012. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730. ACM.

X. Liu, S. Zhang, F. Wei, and M. Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 359–367.

K. Morik, P. Brockhausen, and T. Joachims. 1999. Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring. In *ICML*, volume 99, pages 268–277.

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

N. Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

A. Ritter, S. Clark, Mausam, and O. Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proc. of Empirical Methods for Natural Language Processing (EMNLP)*, Edinburgh, UK.

E. F. T. K. Sang and F. D. Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.