ResToRinG CaPitaLiZaTion in #TweeTs

Kamel Nebhi
LATL-CUI
University of Geneva
7, route de Drize
1227 Carouge, Switzerland
kamel.nebhi@unige.ch

Kalina Bontcheva
University of Sheffield
211 Portobello
Sheffield S1 4DP, UK
K.Bontcheva@dcs.shef.ac.uk

Genevieve Gorrell
University of Sheffield
211 Portobello
Sheffield S1 4DP, UK
G.Gorrell@dcs.shef.ac.uk

ABSTRACT

The rapid proliferation of microblogs such as Twitter has resulted in a vast quantity of written text becoming available that contains interesting information for NLP tasks. However, the noise level in tweets is so high that standard NLP tools perform poorly. In this paper, we present a statistical truecaser for tweets using a 3-gram language model built with truecased newswire texts and tweets. Our truecasing method shows an improvement in named entity recognition and part-of-speech tagging tasks.

1. INTRODUCTION

The growing popularity of Twitter is resulting in the creation of a large number of short messages (tweets) that contain interesting information for several NLP tasks. However, traditional NLP tools have been shown to perform poorly on tweets due to their lack of standardization [7, 8, 9, 19]. The language of social media text is colloquial, and contains a high proportion of mispellings, insertions, neologisms, jargon and non standard capitalization [1].

	Tweets
1	the one and only MARK MEISMER is teaching TONIGHT At EDGe at 830-1030 !
2	ThAnK gOd ItS fRiDaY !!
3	HATE IT WHEN THEY KNOW I'M PAGAN AND INVITE ME . ANGER .

Table 1: Examples of noisy capitalization in tweets.

Twitter users frequently use capitalization as emphasis. For example, in table 1 the first tweet shows that the user has entirely capitalized the word *TONIGHT* and the first letter of a series of words including *At*. Tweet 2 shows that the user has randomly capitalized letters in words such as *ThAnK* or *fRiDaY*. In the last example, every word of the tweet is in uppercase.

In this paper, we show the overall impact of tweet truecasing in Named Entity Recognition (NER) and part-of-speech (PoS) tagging. We demonstrate that our statistical truecaser, which uses a 3-gram language model, can improve NER and PoS tagging accuracy.

The paper is divided as follows. In section 2 we describe work related to case restoration. Then, we present our approach in section 3. Next, in section 4, we show the performance of our 3-gram capitalizer and the benefit of truecasing in NER. Finally, we summarize the paper.

2. RELATED WORK

The case restoration task, also known as truecasing [15], is the process of recovering case information for texts in lowercase. In addition to improving the readability of texts, truecasing is useful for several NLP tasks.

Recently, truecasing has been explored with various methods for statistical machine translation or speech recognition, but has received much less attention in social media normalization.

[15] view truecasing as a lexical ambiguity resolution problem, where several versions of a word happen to have different surface forms. They use an approach based on 3-gram language models estimated from a corpus with case information. [22] exploits bilingual information to create a probabilistic capitalization model using Conditional Random Fields.

[13] presents a purely text-based *n-gram* language model for punctuation and capitalization restoration in order to improve automatic speech transcripts. [1] uses machine-learning techniques to restore punctuation and case in English text. They achieved the best results using a variety of lexical and contextual features, as well as iterative retagging.

[14] creates a normalization pipeline for tweets including a truecaser based on a 3-gram language model. However, this approach has not yet been evaluated. Thus, the impact of recasing in NLP tasks such as NER for social media was not shown. [19] create a capitalization classifier which predicts whether or not a tweet is informatively capitalized. This feature was then used to improve their statistical NER model for microblogs.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2743039.

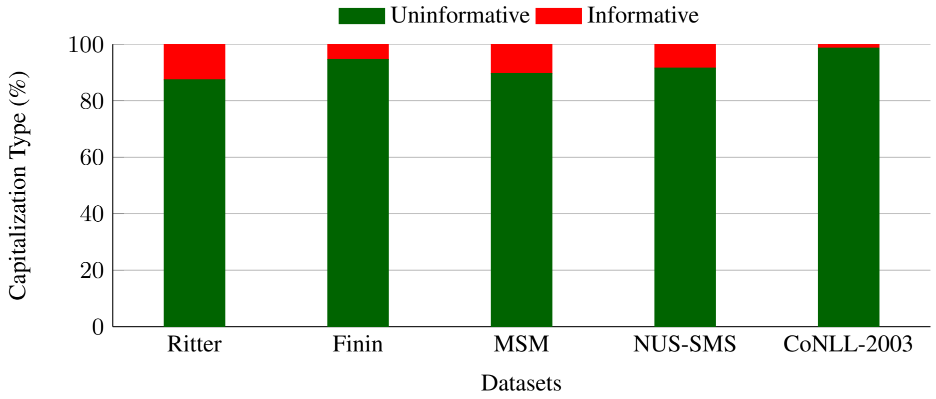

Figure 1: Casing distribution in 3 Twitter datasets, NUS-SMS corpus and CoNLL 2003 test data.

3. APPROACH

3.1 Tweet casing classification

Our truecasing strategy is to determine if the tweet capitalization is *informative* or *uninformative*. To achieve this, we build a classifier that can effectively predict whether or not a tweet is informatively capitalized.

For learning, we train an SVM classifier using these features:

- the fraction of words in the tweet which are capitalized,
- the fraction which appear in a dictionary of frequently lowercase/capitalized words but are not lowercase/capitalized in the tweet,
- the fraction of lowercase/capitalized proper nouns,
- the number of times the word ‘I’ appears in lowercase and whether or not the first word in the tweet is capitalized.

We have used 800 tweets manually labelled as having either *informative* or *uninformative* capitalization. We train the classifier using these 800 tweets and we compute the average 4-fold cross-validation results.

	NB	MaxEnt	SVM
Accuracy	71.25	91.71	94.91

Table 2: Average four-fold cross-validation accuracies in percent.

Table 2 shows that the SVM algorithm performs better than the Naïve Bayes and Maximum Entropy approaches, and the accuracy achieved is around 95%.

3.2 Casing distribution

Using our capitalization classifier, we have analysed the casing distribution in 3 Twitter NER datasets: Ritter [19], Finin [9] and MSM [3]; the NUS-SMS corpus [4] and the CoNLL 2003 testing data [21].

Figure 1 shows that the Ritter and MSM datasets contain around 10 to 12% noisy capitalization. The Finin corpus seems to be a better quality corpus with 5% *uninformative* capitalization.

NUS-SMS corpus contains less than 9% noisy capitalization. Finally, the CoNLL 2003 test data contains less than 2% wrong capitalization[1].

3.3 Data set

To build our truecaser, we train a 3-gram language model using a portion of the English Gigaword Corpus (Fifth Edition), provided by the Linguistic Data Consortium [18]. It consists of newswire texts covering the 24-month period of January 2009 through December 2010 from the following news agencies:

- Agence France-Presse, English Service
- New York Times Newswire Service
- Washington Post/Bloomberg Newswire Service

Additionally, we have used our capitalization classifier to build a corpus of 200,000 tweets containing *informative* capitalization. We have included these tweets in order to improve our language model.

Altogether, the corpus comprises about 2 billion words.

3.4 Architecture

Our truecasing approach used the HMM-based tagger included in the *disambig* tool from the SRILM toolkit [20]. Figure 2 describes the tweet truecasing pipeline. In the first step, we build a recasing model by training a language model on truecased English text (see subsection 3.3). Next, we create a *Map* that contains a list of all the alternative ways that each word can be capitalized. This map lists a lowercased word as the key and associates it with all of the alternate capitalization of that word. This *Map* contains 1.1 million of entries and will be used in our truecaser.

The truecaser is built on GATE [6] and consists of a set of processing resources executed in a pipeline over a corpus of tweets. The pipeline consists of 3 parts:

- Statistical capitalization classifier (categorizes each well and badly capitalized tweet);
- LowerCase sentence step (each badly capitalized sentence is lowercased);
- HMM-based tagger that uses 3-gram language model to compute the probabilities of the most likely case.

[1] The wrong capitalization in the CoNLL 2003 test data is essentially due to the headlines.

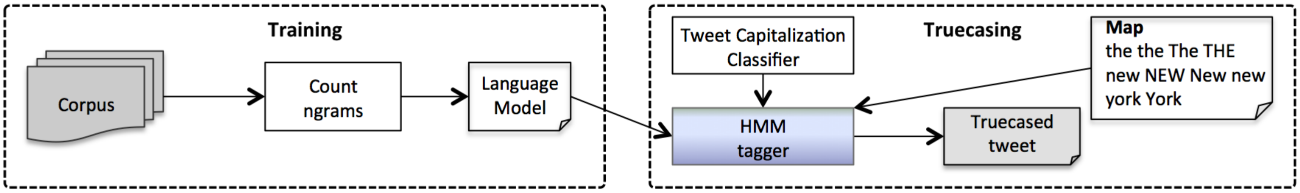

Figure 2: Tweets Truecasing Pipeline.

Corpus	Baseline (lc)	Stanford Truecaser	LM 3-gram (no tweets)	LM 3-gram	LM 4-gram	LM 5-gram
	BLEU scores (%)					
AFP English	73.11	89.15	92.07	94.13	94.19	94.20
Ritter corpus	73.14	72.17	74.45	78.36	78.38	78.39

Table 3: LM Truecaser (with tweet messages and without) vs Stanford Truecaser.

4. EXPERIMENTS

4.1 Truecaser evaluation

We include the lowercase (lc) version as a baseline and we compare our tool to the Stanford Truecaser [16], which is implemented with a discriminative model using CRF sequence tagger [10].

We use BLEU score [17] to measure truecasing performance. BLEU computes the similarity between two sentences using a n-gram statistics, and is widely-used in machine translation evaluation. A set of parallel corpora, consisting of 55,000 sentences from the Agence France-Presse (AFP) corpus and 2,400 English tweets [19] and their normalized references is used as a gold standard.

The performance comparisons between our n-gram truecaser and the lowercase baseline are shown in table 3. The BLEU score shows that the capitalization n-gram model performs substantially better than the baseline. The baseline achieved a BLEU score of 73.11% on the AFP corpus and 73.14% on the Ritter dataset. Our 3-gram language model improves truecasing effectiveness; BLEU score rises to 94.13% on the AFP corpus and 78.36% on Ritter.

The Stanford Truecaser performs better than the baseline with a BLEU score of 89.15% on the AFP dataset, but when applied on tweets it achieved worse results with a BLEU score of 72.17%.

Our 3-gram language model without tweets obtains 92.07% on the AFP corpus and 74.45% on Ritter. Adding tweets to the language model improves BLEU score; 94.13% compared with 78.36%.

Our 4-gram language model achieved 94.19% on the AFP English corpus and 78.38% on the Ritter dataset. The 5-gram language model achieved 94.20% on the AFP English corpus and 78.39% on Ritter. Therefore, increasing the n-gram order does not help nearly as much so we choose to use the 3-gram language model as it is faster.

The 3-gram approach resolves ambiguous cases. For example, with a unigram model “new york” will be recapitalized "new York" because *new* is almost always lowercased. The 3-gram approach takes into account the context and when *new* is followed by *York*, it is almost always capitalized.

4.2 Truecaser output

We have analyzed the differences between results produced by the Stanford Truecaser and our LM Truecaser. Examples (1) and (2) show original microblog messages, including a variety of miscapitalization, and the truecased text (by Stanford Truecaser and our LM Truecaser).

(1) ***Original Tweet:*** Time Warner Cable Boycotting Epix Movie Channel Because It Did A Deal With Netflix
Stanford Truecaser output: Time Warner cable boycotting EPIX movie channel because it did a deal with Netflix
LM Truecaser output: Time Warner Cable boycotting Epix movie channel because it did a deal with Netflix

(2) ***Original Tweet:*** Foooooootball game toooonight then auburn game tomorrow war damn eagle
Stanford Truecaser output: foooooootball game toooonight then Auburn Game Tomorrow War Damn Eagle
LM Truecaser output: foooooootball game toooonight then Auburn game tomorrow war damn Eagle

As illustrated, not all errors are corrected, and our LM Truecaser performs better than the Stanford Truecaser. It also appears that the Stanford Truecaser introduces noisy capitalization.

4.3 Application: Named Entity Recognition

In our experiment on NER for tweets, we have integrated our truecaser in TwitIE[2] [2] and Stanford NER [10].

We evaluate our system on the Ritter and MSM datasets, which contain around 10% to 12% *uninformative* capitalization.

In table 4, TwitIE without truecasing achieves a traditional F-Measure of 47.65% on the Ritter corpus and 66.71% on the MSM dataset. Adding truecasing does not improve accuracy; the system obtains an F-Measure of 47.63% on Ritter and 66.64% on MSM. This result is explained by the fact that TwitIE doesn’t actually use case information when detecting named entities.

Stanford NER without truecasing achieves a traditional F-Measure of 47.34% on Ritter and 73.25% on MSM. Adding truecasing improves extraction effectiveness, and the system obtains an F-Measure of 48.94% on Ritter and 74.64%.

4.4 Application: PoS Tagging

Part-of-speech tagging is necessary for many tasks such as named entity recognition and linking. However, microblog content is difficult to part-of-speech tag as it is noisy and contains linguistic er-

[2] TwitIE is a customisation of ANNIE [5] for microblog content.

	Ritter			MSM		
System	P	R	F1	P	R	F1
TwitIE	46.94	48.38	47.65	68.87	64.69	66.71
TwitIE_LM_TC	46.90	48.38	47.63	68.80	64.62	66.64
StanNER	52.31	43.23	47.34	76.08	70.62	73.25
StanNER_LM_TC	51.81	46.37	48.94	76.62	72.76	74.64

Table 4: Named Entity Recognition performance with truecasing and without on Ritter and MSM datasets. Experiments using TwitIE and Stanford NER (StanNER) systems.

rors. For the last few years, specific taggers [12, 19, 8] have been developed to handle these issues.

For the experiment, we have used two PoS-labeled microblog datasets: a part of the T-Pos corpus introduced by [19] and a part of the DCU dataset [11].

To measure the impact of truecasing on PoS tagging, we have integrated our truecaser in Derczynski et al's tagger [8].

Table 5 shows the benefit of truecasing, giving 89.73% token accuracy on Ritter corpus (21.54% for sentences) and 90.27% token accuracy on MSM corpus (37.60% for sentences).

	T-Pos		DCU	
Tagger	Token	Sentence	Token	Sentence
Derczynski et al., 2013	88.69	20.34	89.37	36.80
Derczynski_TC	89.73	21.54	90.27	37.60

Table 5: Performance (in %) of PoS tagging with truecasing and without on T-Pos and DCU datasets.

5. CONCLUSION

In this paper, we have discussed tweet truecasing, the process of restoring case information in social media messages. We have presented a truecasing method using an n-gram language model built with truecased newswire texts and tweets. Our results suggests that our capitalization 3-gram model performs substantially better than the Stanford Truecaser. Our evaluation shows an improvement on a named entity recognition task. Truecasing also improves the result on a part-of-speech tagging task.

A natural direction in which to continue this research is to adapt this process to languages other than English and also to apply our truecaser to other tasks such as named entity linking.

6. ACKNOWLEDGMENTS

This work was carried out during the first author's stay at Dept. Computer Science, University of Sheffield, funded by the Swiss National Science Foundation (grant P1GEP1_151615).

7. REFERENCES

[1] T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. Wang. How noisy social media text, how diffrnt social media sources? In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, 2013.

[2] K. Bontcheva, L. Derczynski, A. Funk, M. A. Greenwood, D. Maynard, and N. Aswani. Twitie: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics*, 2013.

[3] A. E. Cano, M. Rowe, M. Stankovic, and A.-S. Dadzie, editors. *Proceedings of the Concept Extraction Challenge at the Workshop on 'Making Sense of Microposts', Rio de Janeiro, Brazil, May 13, 2013*. CEUR-WS.org, 2013.

[4] T. Chen and M.-Y. Kan. Creating a live, public short message service corpus: The nus sms corpus. *CoRR*, 2011.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[6] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6)*. 2011.

[7] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. Analysis of named entity recognition and linking for tweets. *CoRR*, abs/1410.7182, 2014.

[8] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, 2013.

[9] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. 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. Association for Computational Linguistics, 2010.

[10] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, 2005.

[11] J. Foster, Ö. Çetinoglu, J. Wagner, J. Le Roux, S. Hogan, J. Nivre, D. Hogan, and J. Van Genabith. # hardtoparse: Pos tagging and parsing the twitterverse. In *AAAI 2011 Workshop on Analyzing Microtext*, pages 20–25, 2011.

[12] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. 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. Association for Computational Linguistics, 2011.

[13] A. Gravano, M. Jansche, and M. Bacchiani. Restoring punctuation and capitalization in transcribed speech. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4741–4744. IEEE, 2009.

[14] M. Kaufmann and J. Kalita. Syntactic normalization of twitter messages. In *International conference on natural language processing, Kharagpur, India*, 2010.

[15] L. V. Lita, A. Ittycheriah, S. Roukos, and N. Kambhatla. truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics, 2003.

[16] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318. Association for Computational Linguistics, 2002.

[18] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. English Gigaword Fifth Edition, 2011.

[19] A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[20] A. Stolcke. Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, 2002.

[21] E. F. Tjong Kim Sang and F. De Meulder. 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*, CONLL '03, pages 142–147. Association for Computational Linguistics, 2003.

[22] W. Wang, K. Knight, and D. Marcu. Capitalizing machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 2006.