An Evaluation of POS Tagging for Tweets Using HMM Modelling

Parma Nand **Rivindu Perera**

School of Computer and Mathematical Sciences
Auckland University of Technology,
Auckland, New Zealand,
Email: {pnand, rperara}@aut.ac.nz

Abstract

Recently there has been an increased demand for natural language processing tools that work well on unstructured and noisy texts such as texts from Twitter messages. It has been shown that tools developed for structured texts, do not work well when used on unstructured texts hence necessitates considerable customization and re-training for the tools to be able to achieve the same accuracy on unstructured texts. This paper presents the results of testing a HMM (Hidden Markov Model) based POS (Part-Of-Speech) tagger customized for unstructured texts.

The tagger was trained on Tweeter messages on existing publicly available data and customized for abbreviations and named entities common in Tweets. We evaluated the tagger firstly training and testing on the same source corpus and later did cross-validation testing by training on one Twitter corpus and testing on a different Twitter corpus. We also did similar experiments with the datasets using a CRF (Conditional Random Frequency) based state-of-the-art POS tagger customized for Tweet messages.

The results show that the CRF-based POS tagger from GATE performed slightly better compared to the HMM model at token level, however at the sentence level the performances were approximately the same. An even more intriguing result was that the cross-validation experiments showed that both the tagger's results deteriorated by approximately 25% at the token level and a massive 80% at the sentence level. This suggests vast differences between the two Tweet corpora used and emphasizes the importance of recall values for NLP systems. A detailed analysis of this deterioration is presented and the HMM trained model together with the data has also been made available for research purposes.

Keywords: Social Media, HMM POS Tagger, Twitter, Machine Learning, POS Tagging

1 Introduction

In the last five years, there has been a significant shift in the way we communicate on the internet. Instead of structured texts, there has been a shift towards loosely structured, short interactive messages. Although, this initially started with text messaging on mobile phones which had a limitation of 140 characters, it has since also proliferated into online communication on popular sites such as Facebook, Twitter, Blogs and Flickr. With such an increase in communication micro-blogging type texts, there has been an increase in demand for appropriate text processing tools for various purposes such as business intelligence and security. Extracting information from such blogs is one of the hardest problems in NLP because of their structure and switching between the type of conversation from one-to-one, multi-party and broadcast messages. NLP methods that work well for longer texts (e.g. named entity recognition, topic identification) have been found to work poorly on blogs and tweets. This has created a need to either adapt existing methods for use with microblog content or find new methods that work well in the specialised domain of micro-blog texts. In response to this need, there has been a flurry of research in recent times from the linguistic point of view in trying to understand the structure of micro-blogging texts, eg., (Monojit Choudhury, Rahul Saraf, Vijit Jain, Sudeshna Sarkar et al, 2007; Cooper et al, 2005; Finin et al, 2010), as well as computational viewpoint in trying to extract information from such texts (Gimpel et al, 2011; Ritter et al, 2010; Barbosa and Feng, 2010; Soderland et al, 1999).

Many of the commonly used IE (Information Extraction) implementations such as Lingpipe[1] and AnnieLingpipe[2] depend on POS tagging in order to perform the downstream tasks, hence this is a crucial step for accuracy in Information Extraction. Initial attempts at POS tagging were done using deterministic, rule-based techniques and some attempts such as (Greene and Rubin, 1971) achieved accuracies as high as 77%. However the inherent difficulties with deterministic techniques such as rule base maintenance and limitations on transportability meant a shift towards probabilistic or stochastic techniques resulting in most of the recent works primarily based on probabilistic techniques with some incorporation of rules. A necessary component of stochastic techniques is supervised learning, which requires annotated training data. Creation of such data is both expensive and time consuming.

Although there are now several sources of accurate POS annotated corpora available in the structured text genre (eg. Penn Tree Bank, Brown Corpus, and MedPost), there is still a dearth of tagged corpora for unstructured texts such as micro-blogs and tweets. Our search for publicly available POS tagged dataset for micro-blogging type texts yielded the following three sources.

[1] *http://alias-i.com/lingpipe.*

[2] *http://gate.ac.uk/ie/annie.html*

- The T-POS dataset (Ritter et al, 2011) consists of 12K tokens from Twitter messages. The corpus uses a set of 38 tags from Penn Treebank (PTB) with additional 4 tags specific to Twitter messages.
- The DCU dataset (Foster et al, 2011) consists of 14K tokens from Twitter messages. This dataset also uses PTB tags, however the additional Twitter specific tags are slightly different to the T-POS dataset.
- The ARK dataset (Gimpel et al, 2011) consists of 39K tokens from Twitter messages. The corpus uses a conflated set of 20 tags from PTB with additional of 5 Twitter specific tags.

Each of the datasets described above has been used in POS tagging experiments and report accuracies of up to 92% using various forms of discriminative models. Gimpel et al (Gimpel et al, 2011) report an accuracy of 92.8% with the ARK dataset using a Conditional Random Field (CRF) estimator. Derczynski et al (Leon Derczynski, Alan Ritter, 2013) again use a CRF estimator on both the T-POS and the DCU datasets and report accuracies as high as 88.7%. Although it is generally accepted that discriminative models (eg. CRF) models perform better than generative (eg. HMM) models (Sutton and McCallum, 2010), generative models by the virtue of their design have some key advantages compared to discriminative models.

One of the key advantages that is pertinent to micro-blogging data, is that generative models are able to better handle datasets which are only partially labelled or completely unlabelled. Generative models require the computation of joint probability distribution of labels and words. The computation for the words does not require labelled data, hence, the probability distribution of words can take advantage of large amounts of unlabelled data for initial training as well as "live" data for real time systems. Secondly, in some cases, as demonstrated by Ng and Jordan (Andrew Y. Ng, 2001), generative models perform better when the input model has a smoothing effect on the features. Their results show that the advantage of generative models is even more pronounced when the dataset is small as is currently the case for labelled micro-blogging data. A third advantage of generative models is that it's training time is insignificant compared to discriminative models, hence has an advantage in real time applications where progressive learning is required.

On the other hand discriminative models have better generalization performance when training data is abundant with the ability to account for more global features compared to a generative model. This gives discriminative models the ability to model features from any arbitrary part of a sentence, not necessarily in a linear fashion. This freedom enables it to model a much larger set of features, however it also exposes the model to the risk of overfitting the training which leads to poor generalization on unseen data. For a detailed discussion and comparison of various probabilistic models see Kalinger(Roman Klinger, Katrin Tomanek, 2007).

This paper investigates the generalization ability of two discriminative, pre-trained Twitter tagging systems and evaluates them against a generative model using three Twitter datasets, T-POS, DCU and ARK. We used the Hidden Markov Model (HMM), the basic implementation of which as adapted from LingPipe[3]. The HMM tagger was used to train on a subset of the tweet data in each of the datasets and the results were compared against two discriminative models, a Maximum Entropy model as implemented in the Stanford POS Tagger[4] and a highly customized CRF model implemented as an extension in GATE[5]. We also did cross-dataset validation and observed that the performance deteriorates by approximately 20% in all the models tested. In this paper we present an evaluation of the comparative performance of the two classes of taggers as well as an analysis of the deterioration of the results in the cross validation experiment.

In summary this paper make the following contributions:

- Makes the source code and jar publicly available[6] for further research or to use the HMM tagger for tagging Tweet messages.
- Presents the results of direct comparison between three types of POS taggers trained and tested on Tweet messages.
- Presents an evaluation of cross-dataset generalizability of three classes of tagging models.
- Presents a detailed analysis of the error contribution for each class of tagging model.

2 Characteristics of Tweet Data

There is much linguistic noise in micro-blogging texts as a result of the ways in which micro-blogs are written. The noise arise from different language usage characteristics, hence different strategies are required to account for them. The following are a list of the characteristics with a discussion of the challenges resulting due to the linguistic noise.

Word Capitalization Use of capitalization in micro-blogs is inconsistent. In formal texts, capitalization is used as a key feature for recognizing proper nouns (tags NNP and NNPS), however in micro-blogs capital letters are used for several other purposes such as emphasis(eg. ".tomorow YOU will need to do that") and highlighting (eg. "lease do me a favor and POST CHAPTER 15"). Micro-blog texts also contain a plethora of capitalization due to typos. Apart from these, noise is also introduced by a lack of capitalization of nouns which should be capitalized. Various techniques such as use of name lexicon lists (Leon Derczynski, Alan Ritter, 2013) and the use of a trained classifier to recognize a valid capitalization (Ritter et al, 2011) have been used to account for the noise due to capitalization.

Word Variations Spelling variations could be unintentional, since Microblog texts rarely get proofread, or intentional, for the purpose of compressing long words, eg. use of *tmro* and *2moro* for tomorrow. Word compressions can take either a graphemic or a phonemic form (Monojit Choudhury, Rahul Saraf, Vijit Jain, Sudeshna Sarkar et al, 2007). Graphemic forms involve deletion vowels (eg. "msg"), deletion of repeated characters (eg. "tomorow" for "tomorrow") and truncation (eg. "tom" for "tomorrow"). Phonemic

[3] *http://alias-i.com/lingpipe.*

[4] *http://nlp.stanford.edu/software/tagger.shtml*

[5] *http//gate.ac.uk/wiki/twitter-postagger.html*

[6] *http://staff.elena.aut.ac.nz/Parma-Nand/projects/TaggerDownload.html*

forms involve substitution of a shorter set of characters to represent the same phoneme, eg. “2” in “2moro”. Choudhury et. al report decoding such spelling variations with an 80% success using a HMM model.

Multi Word Abbreviation Frequently, multiple words are abbreviated as single word abbreviations, eg. “lol” for “laugh out loud” and “tgif” for “thank god its Friday”. These abbreviations can only be identified by use of lexicons.

Slangs Slangs such as “gonna” used for “going to” is common since it reduces the size of the sentence. Lexicons with word-to-phrase maps have been used with varying levels of success to account for such use of slangs (Gimpel et al, 2011; Derczynski et al, 2013).

Word Omission Frequently used function words such as articles and subject nouns are omitted. For example, “I went to town in a car” may be written as “went town in car”. This category of noise is easily handled by training sequence based probabilistic models and both CRF and HMM models perform well with this type of noise.

3 Data Sets Used

Currently, there are 3 sets of tagged datasets on tweet texts, two of them (TPOS and DCU) use similar PTB tag set, while the third one (ARK) uses a much smaller subset of 25 tags. The list of tags are shown in Figures 1 and 2 for TPOS/DCU and ARK respectively.

$, ”, (,), ,, ., :, ADVP, CC, CD, DT, EX, FW, HT, IN, JJ, JJR, JJS, MD, NN, NNP, NNPS, NNS, PDT, POS, PRP, PRP$, RB, RBR, RBS, RP, RT, TO, UH, URL, USR, VB, VBD, VBG, VBN, VBP, VBZ, WDT, WP, WRB.

Figure 1: Alphabetical list of tags from Penn Treebank used in the TPOS and DCU dataset

!, #, $, &, ,, @, A, D, E, G, L, M, N, O, P, R, S, T, U, V, X, Y, Z, ˆ, .

Figure 2: Alphabetical list of tags used in the ARK dataset.t

The PTB data sets contains the following 3 Twitter specific tags

1. **URL** - url, eg. http://www.illocuti oninc.com/
2. **HT** - topic, eg. #newyear
3. **USR** - a tweet username, eg. @catlovesit

The ARK dataset contains the Twitter specific tags from PTB and in addition introduces two additional tags for continuation of a tweet ( ) and emoticon (eg. :-)). It uses the following symbols for the 5 additional Twitter specific tags:

1. **U** - url, eg. http://www.illocuti oninc.com/
2. **#** - topic, eg. #newyear
3. **@** - a tweet username, eg. @catlovesit
4. **E** - emoticon, eg. :-)
5. - continuation of a tweet, always preceded by

The ARK dataset, introduced in (Gimpel et al, 2011), used 17 annotators to tag a total of 1827 English Tweets containing approximately 26,000 tokens. The paper reports an inter annotator-agreement rate of 92.2%. This data was then used to train a CRF tagger model with additional 5 features to handle various types of linguistic noise such as orthography, phonetics and capitalization. The authors report an accuracy of 89.37% compared to 85.85% for a retrained Stanford tagger.

The TPOS dataset based on PTB was first introduced in (Ritter et al, 2011), contains 12K tokens from Tweet messages. The authors report a tagging accuracy of 88.3% with a CRF tagger model trained on a mixture of TPOS (12K), IRC[7] (40K) and PTB (50K). The accuracy reported from this study was 88.3% compared to 80.01% for the Stanford tagger.

The DCU dataset, again based on PTB, introduced in Foster (Foster et al, 2011) contains 269 annotated sentences which had a reported inter-annotator agreement rate of 95.8%. The authors use the dataset to train and test a Support Vector Machine tagger and report an accuracy of 84.4% compared to an accuracy of 96.3% for Wall Street Journal data. This paper focusses on parsing, hence no tagging specific up-training is done to account for the linguistic noise, however the public availability of the tagged data is extremely useful for Twitter tagging research.

4 Experimental Setup

The testing was conducted using the 3 different sets of Tweet data described in section 3 so that cross validation performance could also be determined. We used the Stanford tagger (Stanford) and the enhanced Stanford tagger (Gate), both shipped with the GATE package[8]. The Stanford tagger has been shown to exceed accuracies of over 90% (Leon Derczynski, Alan Ritter, 2013) , hence is considered to be state-of-the-art. We used the Stanford tagger as a benchmark, and did cross validation tests on its enhanced version (Gate tagger) across the three datasets and also did comparison tests against a newly introduced HMM model.

For the Stanford and Gate taggers, we used the pre-trained standard models (as opposed to faster models with lower accuracy) named *english-twitter.model* and *gate-EN-twitter.model* respectively. These two models were tested against a twitter trained HMM model, modified from the basic implementation in LingPipe[9]. A final evaluation was done to compare the training times between a CRF and a HMM model as implemented in LingPipe. This evaluation was done without the implementation of any additional features in both the CRF and the HMM model for the purpose of comparing the computational cost for the two models.

Table 1 gives the details of the data splits between training and testing sets used in the evaluation experiment. The training and testing splits for T-POS and DCU dataset is the same as that used for training the pre-trained GATE models reported in Derczynski et al (Leon Derczynski, Alan Ritter, 2013).

[7] Chat data from (Forsythand and Martell, 2007)

[8] *http://gate.ac.uk/wiki/twitter-postagger.html*

[9] *http://alias-i.com/lingpipe*

Dataset	Sentences	Tokens
T-POS-train	551	10.6K
T-POS-test	118	2.2K
DCU-train	269	2.9K
DCU-test	250	2.8K
ARK-train	1827	26.6K
ARK-test	547	7.7K

Table 1: Dataset Details

The first test on the HMM model was done using the ARK data set split, into training and testing set as in Gimpel et. al (Gimpel et al, 2011). (shown on the last 2 rows of Table 1). The HMM model was initially tested for n-gram (number of previous tokens used for emissions) for values ranging from 1 to 10. An n-gram value of 5 gave us the best performance, and we fixed the HMM model at this value for all the other tests.

The HMM model was first trained by varying the amounts of training data and tested on the ARK-test data which consists of 547 individual Tweets. The accuracy very quickly reached 80% at about 300 Tweets after which the the increase was slow up to a maximum value of 87% for the total amount of training data of 1827 Tweets. This compares very well with the result reported in (Gimpel et al, 2011) for a substantially feature engineered CRF model which obtained a value of 89.37%. As a comparison, the Stanford tagger had an accuracy of 85.85% on the same training and test data.

The TPOS and the DCU data sets use the PTB tagset and hence the results for these two dataset are directly comparable. The ARK data set contains a smaller subset of 25 tags by conflating some of the PTB tages. A model trained on the subset dataset cannot be cross validated on a superset dataset, however the opposite is possible by mapping the superset onto the subset tagset. Hence, the tagging results from the superset trained systems was cross validated on the ARK dataset by mapping the superset tags to the subset tags, for example all forms of verbs (VB, VBD, VBG, VBN VBP VBZ) were mapped to a single tag, V, in the ARK dataset.

Table 2 shows the results for the cross validation tests of the four models tested on three datasets. The pre-trained Stanford and Gate taggers tested on the TPOS-test and DCU-test data sets achieved relatively high accuracies, close to the reported values in (Leon Derczynski, Alan Ritter, 2013). For example, the Gate tagger achieved a token accuracy value of 93.5% on the TPOS-test data and 89.4% on the DCU-test data. The corresponding token accuracy values for the LingPipe tagger was 82.8% for TPOS-test and 82.7% for the DCU-test data. As another comparison, the dataset were also used to train and test a CRF model from LingPipe. This model which was devoid of any domain specific feature engineering gave much lower accuracy on all data sets, shown in the last row of Table 2. The LingPipe(CRF) model was tested mainly for the purpose of comparing the training time rather than for accuracy. As an indication the training time for the CRF model on the TPOS-train data set was approximately 8 hours for 200 epochs and had to be left overnight for 1000 epochs. This compares with less than 30 seconds for the HMM model for the same training data. For text processing tasks, each word is a feature. Hence for a discriminative model such as the CRF, there needs to be multiple iterations through the whole feature set in order to be able to find discriminative features for each of the tags. This adds a huge computational cost making it unsuitable for real time systems. On the other hand, a generative model such as HMM, needs to traverse through all the features (words and tags) once and determine the probability distribution of token and word emissions. This can then be easily updated as new features are encountered making it adaptable as new data is encountered making it suitable as a progressive learner.

The three models were then cross validated by training them with TPOS-train data and tested against the ARK-test data. Since the ARK-test data uses a smaller set of tags, the output of the models trained on TPOS-train data were first mapped to the ARK tagset before running the evaluation tests. Intuitively, this was expected to give us an even higher accuracy since the mapping is “downward”. For example, all confusions between VB, VBD, VBG, VBN and VBP from PTB tagset were mapped to V from the ARK tagset, which would be expected to drive up the accuracy since we are evaluating against a more coarse set of tags. The accuracies obtained for this cross validation are shown in Table 3. All the three models being evaluated (Stanford, Gate and Lingpipe) trained on TPOS-train data give very close results for TPOS-test and DCU-test datasets, hence for cross validation, the results of these two accuracies were averaged and then compared with the ARK-test accuracy shown in Table 3. The cross validation results show a drop of token accuracy between 21 and 25 percent, while the sentence level drop is even more substantial with values of approximately 80%. The accuracy difference between the models cannot be attributed to the feature engineering in CRF models since a similar drop was also observed in the HMM model. The details of the confusions is investigated in section 5. The individual tag accuracies in Figure in 3 shows very similar tag-based performance characteristics for both Gate and HMM models. The tags between *ADVP* and *RBR* were low in number (below 10) hence the 0% accuracy for the HMM model. Apart from the *NNPS* (singular proper noun) tag, Gate consistently performs better or equivalent for the rest of the tags. In the case of *NNPS*, Gate implements several feature engineering techniques in order to identify un-capitalized proper nouns, however in the case of the TPOS-test data, this was counterintuitive compared to the HMM model which essentially uses capitalization to identify proper nouns. The figures in 4 and 5 show the individual tag performance for the Gate and the HMM models trained on TPOS-train data. Both the models show that there is an average of 20% drop in accuracy for ARK-test data is and this is due to a consistent lower performance across all the tags rather than an aberration pertaining to a subset of tags, which could have been possible in the ARK-test data. A candid analysis of the tagging between TPOS and the ARK data sets did not show any gross tagging differences, hence the performance degradation has to be attributed to differences in higher level data characteristics. This is currently being investigated.

5 Error Analysis of the HMM Model on the TPOS data

The TPOS-test data contains a total of 2291 tokens tagged with 44 tags which consists of 41 PTB tags and additional 3 twitter specific tags. Table 4 shows the confusion distribution for the tags which contributed more than 20% error for the HMM model tested on the TPOS data. The numbers in brackets in the 3

Model	**TPOS-test**		**DCU-test**		**ARK-test**	
	Tok	Sent	Tok	Sent	Tok	Sent
Stanford(MaxENT)(pre-trained)	88.7	20.3	89.4	36.8	69.6	6.4
Gate(CRF)(pre-trained)	93.5	33.8	89.4	37.6	69.5	6.3
LingPipe(HMM)(TPOS-train)	82.8	25.2	82.7	24.8	61.9	4.8
LingPipe(HMM)(ARK-train)	-	-	-	-	86.9	26.4
LingPipe(CRF)(TPOS-train)	69.7	4.4	64.0	4.1	63.2	4.1

Table 2: Percentage Accuracies for the cross validation tests

Model	**TPOS/DCU-test avg.**		**ARK-test**		**%age drop**	
	Tok	Sent	Tok	Sent	Tok	Sent
Stanford(MaxENT) (pre-trained)	89.05	21.8	69.6	6.4	21.8	77.6
Gate(CRF) (pre-trained)	91.45	24.0	69.5	6.3	24.0	82.6
LingPipe(HMM) (TPOS-train)	82.75	25.2	61.9	4.8	25.2	80.8

Table 3: Percentage accuracy drops for cross validation tests for the three types of taggers.

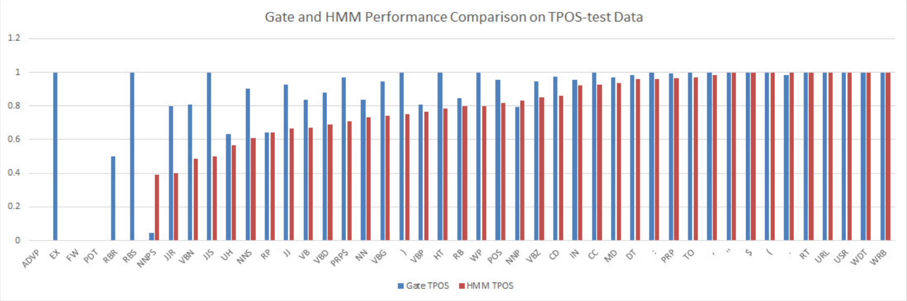

Figure 3: A comparison of the Gate and HMM models' performance for individual tags, sorted in ascending order for ARK data values. The sample used is for a TPOS trained model tested on TPOS-test data.

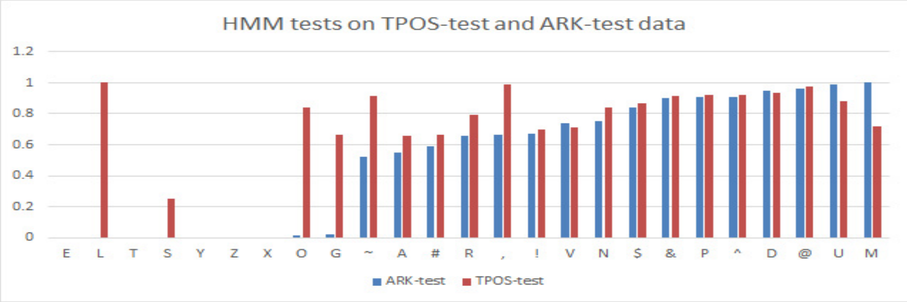

Figure 4: Gate Model's performance on individual tags for TPOS-test and ARK-test Data, sorted in ascending order for ARK data values.

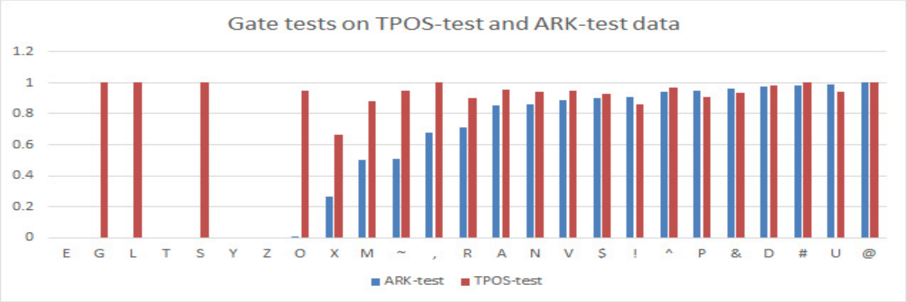

Figure 5: HMM Model's performance on individual tags for TPOS-test and ARK-test Data, sorted in ascending order for ARK data values.

Tag	Total	Correct	Accuracy	Conf-1	Conf-2	Conf-3
JJ	93	53	0.57	NN(19)	NNP(11)	RB(4)
RB	92	74	0.80	NN(7)	IN(4)	JJ(3)
NN	297	239	0.80	JJ(11)	VB(5)	NNS(5)
VBN	18	10	0.56	VBD(4)	NN(2)	JJ(1)
VB	118	85	0.72	VBP(5)	VBD(4)	NNP(4)
VBP	68	49	0.72	VB(11)	RB(2)	NN(2)
VBZ	50	35	0.70	NNS(9)	NNP(4)	VB(1)
VBD	52	40	0.77	NN(3)	VBN(3)	JJ(2)
NNS	49	35	0.71	NN(9)	NNP(3)	RB(1)
NNP	181	130	0.72	NN(24)	JJ(6)	JJ(6)
UH	86	68	0.79	:(4)	RB(3)	NN(3)

Table 4: TPOS-train data trained HMM results for tags with accuracy below 80%. Conf(1) is the tag with highest confusion and Conf(3) is the third lowest.

confusion columns show the number of confused instances for each of the tags. The lowest accuracy was achieved for the *JJ* (adjective) tag with an accuracy of 0.57. The *JJ* tag was confused with *NN* (common noun) 17 times, *NNP* (singular proper noun) 11 times and *RB* (adverb) 4 times. From the rest of the rows in table 4 it can be seen that the *JJ* tag features as the highest number of false positives for the other confused tags as well. Adjectives are most difficult tags to identify as many of the nouns also function as adjectives, as for example, "valuable player" and "Costa Rican group". In these examples the tokens "valuable", "Costa" and "Rican" function as adjectives, however were identified as nouns. Out of the 30 confusions for nouns and pronouns, 60% were in the category of compound noun modifier adjectives, which were identified as either nouns or proper nouns. The rest of the adjectives were of the form where an adjective was used in a position other than a pre-modifier. For example in "...I just felt special...", the token "special" is functioning as an adjective in a syntactical position which is usually a noun in most clauses. Hence, in sequence oriented, probability based models such as HMM, the tagging for adjective will be biased towards nouns. The bias is only corrected if the exact token was present in the training data, which is why probability based models are only as good as the range and extent of the training data. Another category of example which accounted for a high proportion of confused adjectives was the token "long" in "...all year long...". In this case the tokens "all" and "year" are a determiner and noun respectively, hence the adjective is a post modifier of the compound noun. This is another rare syntactical position for an adjective. The majority of adjectives in this category were classified as nouns as the last token of a compound noun is frequently a noun. The *VBN* (past participle) tag is the other tag with accuracy in the 50% range. There were only 18 *VBN* tags in the test data and 4 of these were confused with *VBD* (past tense), which is a tag that can be represented by the same set of words distinguished only by a complex combination of the rest of the tokens in the sentence. The other significant confusion worth mentioning is the confusion of the *NNP* (proper noun) tag confused with *NN* (common noun). This is due to a lax capitalisation in tweet messages. The HMM model used for the testing used capitalisation in addition to the city, corporation and name lexicons from the Stanford implementation to tag proper nouns, hence the 24 confusions out of the 181 were outside these lexicons. Tagging of the *NNP* tags can be easily improved by extending the existing Stanford lexicon lists for specific applications.

6 Concluding Remarks

Social media micro-blogging sites such as Twitter contain vast amounts of information which is extremely current and, as a result, is fast changing. This necessitates text processing tools which are both efficient as well as has the ability to do progressive learning. Since POS tagging is one of the most fundamental tasks for text processing, this paper presented the results of a comparison of two popular POS modelling techniques, the CRF and the HMM models. The state-of-the-art CRF tagger, part of the Gate package, was first tested on the TPOS corpus which was used for its development and then cross-validated with a different corpus, ARK. Both of these corpora was similarly used to test a HMM model, the basic form of which is part of the Lingpipe package. The modified HMM model together with the all the data in the appropriate formats has been made available at (URL removed and can be supplied as supplementary file) for research use.

The token accuracy for the HMM model was found to be 8% below the CRF model, but the sentence accuracy for both the models was very close, approximately 25%. The cross validation results for both the models showed a degradation of approximately 25% for tokens and a very drastic drop of approximately 80% at the sentence level. The degradation in accuracy across the two corpora implies that the two datasets had slightly different characteristics hence a model trained on one set had impaired performance on the other set. The comparative performance of the HMM and the CRF models show that the CRF model is marginally better, however the HMM model learns orders of magnitude faster and is easily adaptable to progressive learning applications. Hence, is better suited to real time applications such as processing live tweets and streaming texts.

References

Andrew Y Ng MIJ (2001) On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes

Barbosa L, Feng J (2010) Robust Sentiment Detection on Twitter from Biased and Noisy Data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pp 36–44

Cooper R, Ali S, Bi C (2005) Extracting Information from Short Messages. In: Montoyo A, Muoz R, Métais E (eds) Natural Language Processing and Information Systems, Lecture Notes in Computer Science, vol 3513, Springer Berlin Heidelberg, pp 388–391, DOI 10.1007/11428817_44, URL http://dx.doi.org/10.1007/11428817_44

Derczynski L, Maynard D, Aswani N, Bontcheva K (2013) Microblog-genre noise and impact on semantic annotation accuracy. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media - HT '13, ACM Press, New York, New York, USA, pp 21–30, DOI 10.1145/2481492.2481495

Finin T, Murnane W, Karandikar A, Keller N, Martineau J, Dredze M (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, Association for Computational Linguistics, Stroudsburg, PA, USA, CSLDAMT '10, pp 80–88

Forsythand EN, Martell CH (2007) Lexical and Discourse Analysis of Online Chat Dialog. In: International Conference on Semantic Computing (ICSC 2007), IEEE, pp 19–26, DOI 10.1109/ICSC.2007.55

Foster J, Çetinoglu O, Wagner J, Le Roux J, Hogan S, Nivre J, Hogan D, Van Genabith J (2011) #hardtoparse: POS Tagging and Parsing the Twitterverse. In: AAAI 2011 Workshop On Analyzing Microtext, pp 20–25, URL http://hal.archives-ouvertes.fr/hal-00702445

Gimpel K, Schneider N, O'Connor B, Das D, Mills D, Eisenstein J, Heilman M, Yogatama D, Flanigan J, Smith NA (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, Stroudsburg, PA, USA, HLT '11, pp 42–47

Greene BB, Rubin GM (1971) Automatic Grammatical Tagging of English. Department of Linguistics, Brown University

Leon Derczynski, Alan Ritter SC (2013) Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. Proceedings of the International Conference on Recent Advances in Natural Language Processing

Monojit Choudhury, Rahul Saraf, Vijit Jain, Sudeshna Sarkar AB, Choudhury M, Saraf R, Jain V, Sarkar S, Basu A (2007) Investigation and modeling of the structure of texting language. In: In Proceedings of the IJCAI Workshop on "Analytics for Noisy Unstructured Text Data, pp 63–70

Ritter A, Cherry C, Dolan B (2010) Unsupervised Modeling of Twitter Conversations. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pp 172–180

Ritter A, Clark S, Mausam, Etzioni O (2011) Named entity recognition in tweets: an experimental study pp 1524–1534

Roman Klinger, Katrin Tomanek RK (2007) Classical Probabilistic Models and Conditional Random Fields

Soderland S, Cardie C, Mooney R (1999) Learning Information Extraction Rules for Semi-structured and Free Text. In: Machine Learning, pp 233–272

Sutton C, McCallum A (2010) An introduction to conditional random fields. arXiv preprint arXiv:10114088