# A HMM POS Tagger for Micro-Blogging Type Texts

Parma Nand, Rivindu Perera and Ramesh Lal

School of Computer and Mathematical Sciences,
Auckland University of Technology,
Auckland 1010, New Zealand
{parma.nand, rivindu.perera, ramesh.lal}@aut.ac.nz

**Abstract.** The high volume of communication via micro-blogging type messages has created an increased demand for text processing tools customised the unstructured text genre. The available text processing tools developed on structured texts has been shown to deteriorate significantly when used on unstructured, micro-blogging type texts. In this paper, we present the results of testing a HMM based POS (Part-Of-Speech) tagging model customized for unstructured texts. We also evaluated the tagger against published CRF based state-of-the-art POS tagging models customized for Tweet messages using three publicly available Tweet corpora. Finally, we did cross-validation tests with both the taggers by training them on one Tweet corpus and testing them on another one.
The results show that the CRF-based POS tagger from GATE performed approximately 8% better compared to the HMM (Hidden Markov Model) model at token level, however at the sentence level the performances were approximately the same. The cross-validation experiments showed that both tagger's results deteriorated by approximately 25% at the token level and a massive 80% at the sentence level. A detailed analysis of this deterioration is presented and the HMM trained model including the data has also been made available for research purposes.
Since HMM training is orders of magnitude faster compared to CRF training, we conclude that the HMM model, despite trailing by about 8% for token accuracy, is still a viable alternative for real time applications which demand rapid as well as progressive learning.

## 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, Flickr and YouTube. With an increase in communication via these micro-blogging sites, there is an increasing demand for information extraction from such semi-structured texts for business intelligence, security, planning and other purposes. However, extracting information from such blogs is one of the most challenging tasks in NLP (Natural Language Processing) because of their unusual structure which may also involve switching between one-to-one and multi-party conversations including 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 an

urgent need to either adapt existing methods for use with microblog content or find new methods that work well in the specialised domain of microblog 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., [14,4,7]), as well as computational viewpoint in trying to extract information from such texts [10,18,3,21].

Information extraction (IE) is the process of identifying some predefined set of concepts from free text. Some of the common tasks in information extraction includes identification of named entities, relations such as temporal and causal, events and identification of other types of information dictated by the application on hand. The process of information extraction can be tackled using a host of approaches. While a lot of works has been done on IE for formal domains such as bio-medical data, newspaper articles, and research articles (eg., [16,2,6,17]), there is limited work on informal domains such as micro-blogs. Some IE approaches can work without any degree of pre-processing, however others require some degree or preprocessing. For example, Cooper [4] reports an approach which does not use any form of pre-processing. This approach uses pattern matching templates resembling the underlying structures of the text data, which can be used to extract various pieces of information directly from Short Message Service (SMS) texts. However, most other approaches rely on some degree of pre-processing of the text before attempting an IE task. Pre-processing might involve full parsing, or more commonly, coarse level tasks such as tokenization and part-of-speech (POS) tagging. For example, Nebhi [15] reports a rule based system working from ontology [1] to extract information such as classes, properties and instances. This system relies on both tokenization and POS tagging (uses Stanford POS tagger) to identify entities from the ontology. It then applies rules for creating the required semantic annotations for classes and their associated properties.

The other commonly used IE implementations such as Lingpipe[2] and AnnieLingpipe[3] 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 [11] achieved accuracies as high as 77%. However the inherent difficulties with deterministic techniques such as rule base maintenance and limited on transportability. This resulted in a shift towards probabilistic or stochastic techniques, hence most of the recent works are primarily based on probabilistic techniques with some incorporation of rules. A necessary component of stochastic techniques is supervised learning, which requires training data. Training data for POS tagging requires existing POS tagged data. This data has to be fully or partially tagged by a human, which is expensive and time consuming.

Although there are now several sources of accurate POS annotated corpora available for 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://wiki.dbpedia.org/Ontology
[2] *http://alias-i.com/lingpipe.*
[3] *http://gate.ac.uk/ie/annie.html*

- The T-POS dataset [19] 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 [9] 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 [10] 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. [10] report an accuracy of 92.8% with the ARK dataset using a Conditional Random Field (CRF) estimator. Derczynski et al. [13] 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) perform better than generative models (eg., HMM) [22], the generative models have some key advantages compared to discriminative models due to their design.

Firstly, the 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 [1], 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. The 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 had discriminative models have better generalization performance when training data is abundant providing 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 data which leads to poor generalization on unseen data. For a detailed discussion and comparison of various probabilistic models see Kalinger [20].

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) as implemented in LingPipe[4] to train on a subset of the tweet data in each of the datasets and compare the results obtained against two discriminative models, a Maximum Entropy model implemented the Stanford POS Tagger[5] and a highly customized CRF model im-

[4] *http://alias-i.com/lingpipe.*
[5] *http://nlp.stanford.edu/software/tagger.shtml*

plemented as an extension in GATE[6]. 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 of our cross-validation experiment.

In summary this paper makes the following contributions:

- Makes publicly available the source code and jar file for further research or to use the HMM tagger for tagging Tweet messages. This can be downloaded from *http://staff.elena.aut.ac.nz/Parma-Nand/TaggerDownload.html*.
- 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 the two classes of tagging models.
- Presents a detailed analysis of the error contribution for each class of tagging model.

## 2 Noise Sources in 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 [13] and the use of a trained classifier to recognize a valid capitalization [19] 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 [14]. Graphemic forms involve deletion vowels (eg. "msg"), deletion of repeated characters (eg., "tomorow" for "tomorrow") and truncation (eg. "tom" for "tomorrow"). Phonemic 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.

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

**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 [10,5].

**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 Micro-blogging Data Sets

In general Tweets, SMS, and micro-blogging texts present a challenge for text processing models because of excessive amounts of linguistic noise. The origin of this linguistic noise is partly historical. Tweeting and some forms of miro-blogging originated from exchange of text messages between phones, which had a 140 character limit with a small keypad causing excessive typing errors. This has remained as part of Tweets and at least some of the micro-blogs even though the activity has largely shifted from small keypad phones to computers and other forms of mobile devices with full keypads. In spite of the informal nature of micro-blog texts and length limitation, there is an enormous amount of information embedded in micro-blogs, in particular in Tweets. The corpus of only one form of micro-blog, Tweets, is already larger than the size of Library of Congress [12] and is growing by gigabytes on a daily basis. Typically Micro-blogging sites have millions of users from around the world with constant activity. This makes the embedded information in them up-to- the-minute current and a good sample representation of the population. Hence information extraction from such a source would invaluable for just about any application.

POS tagging is one of the key pre-processing tasks for any information extraction system. Hence there is a need for accurate and efficient POS tagger for subsequent tasks in the information extraction pipeline. The state-of-the-art taggers available today have been trained and adapted for the newswire genre of texts. Hence they don’t perform as well on out-of-domain texts. Models trained on newswire data have been shown to have error rates up to 10 times higher on texts from the micro-blog genre [5]. This makes it critical that probabilistic taggers are trained and customized on in-domain data before they get used in an NLP pipeline. Currently, there are 3 sets of tagged datasets on tweet texts. Twoof them (TPOS and DCU), use similar PTB tag set, while the third one (ARK) uses a much smaller subset of 25 tags. Figures 1 and 2 show the tags used in TPOS/DCU and ARK datasets.

The PTB data sets contains 3 Twitter specific tags (URL, HT and USR) while the ARC set contains these 3 but in addition includes 2 more. The Tweeter specific tags for both datasets are:

1. **U** - url, eg., http://www.illocuti oninc.com/ (URL in PTB dataset)

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

**Fig. 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, ^, .

**Fig. 2.** Alphabetical list of tags used in the ARK dataset

2. **#** - topic, eg., #newyear (HT in PTB dataset)
3. **@** - a tweet username, eg. @catlovesit (USR in PTB dataset)
4. **E** - emoticon, eg., :-)
5. **~** - continuation of a tweet, always preceded by

The ARK dataset, introduced in [10], 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%. The 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 results show an accuracy of 89.37% compared to 85.85% for a retrained Stanford tagger.

The TPOS dataset based on PTB was first introduced in [19], 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 [9] contains 269 annotated sentences which had a reported inter-annotator agreement rate of 95.8%. In this study the dataset is used 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 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% [13] , 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.

[7] Chat data from [8]

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

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 for both the CRF and the HMM models for the purpose of comparing the computational cost of the two models.

**Table 1.** Dataset Details

| **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 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. [13].

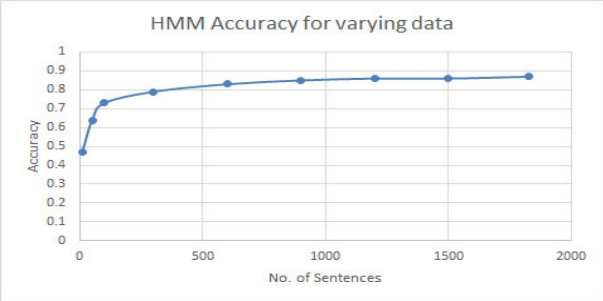


**Fig. 3.** Testing Accuracies for ARK-test dataset for increasing amounts of training data from ARK-train dataset

The initial test on the HMM model was done using the ARK data set by splitting it into training and testing set as in Gimpel et. al [10]. (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

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

performance which was the assigned value used for the HMM model for all subsequent 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 results are shown in Figure 3. The accuracy very quickly reached 80% at about 300 Tweets after which 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 [10] 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.

**Table 2.** Percentage Accuracies for the cross validation tests

| **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 3.** Percentage accuracy drops for cross validation tests for the three types of taggers.

| **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 |

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 [13]. 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 datasets 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 which makes 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 4 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 counter-intuitive compared to the HMM model which essentially uses capitalization to identify proper nouns. The results in Figure in 5 and 6 show the individual tag performance for the Gate and 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.

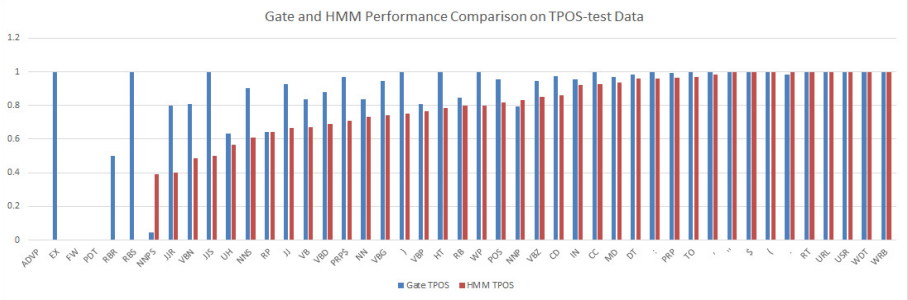


**Fig. 4.** A comparison of the Gate and HMM models' performance for individual tags, sorted in ascending order for ARK data values. The sample used is for a TPOS trained model tested on TPOS-test data.

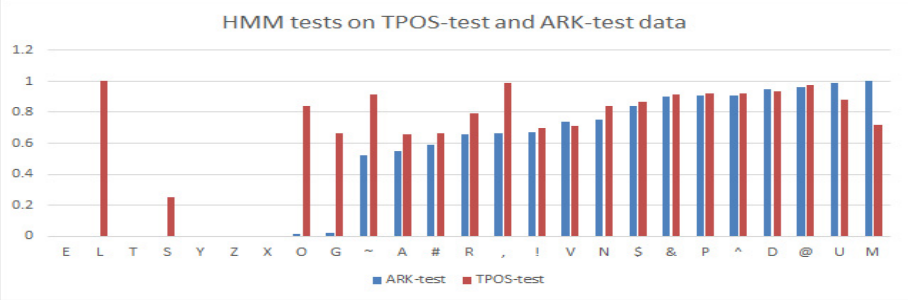


**Fig. 5.** Gate Model's performance on individual tags for TPOS-test and ARK-test Data, sorted in ascending order for ARK data values.

## 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 confusion columns show the number of confused instances for each of the tags. The lowest accuracy was achieved for

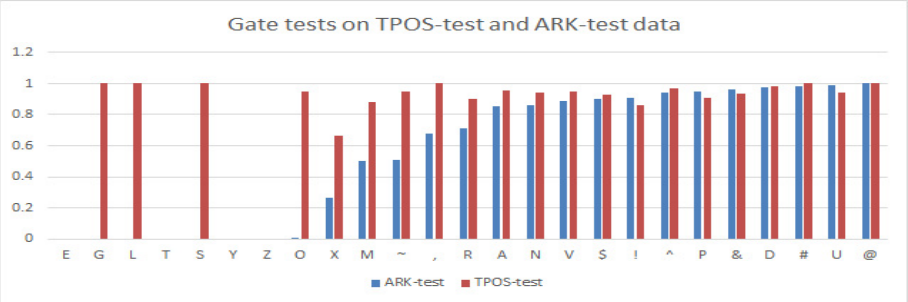


**Fig. 6.** HMM Model's performance on individual tags for TPOS-test and ARK-test Data, sorted in ascending order for ARK data values.

**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.

| 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) |

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 can only be 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 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

This paper presented HMM POS tagger customized for micro-blogging type texts. We also presented the results of comparison with a state-of-the-art CRF tagger. 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

1. Andrew Y. Ng, M.I.J.: On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes (2001)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction for the web. In: IJCAI. vol. 7, pp. 2670–2676 (2007)
3. Barbosa, L., Feng, J.: Robust Sentiment Detection on Twitter from Biased and Noisy Data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 36–44. COLING ’10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
4. Cooper, R., Ali, S., Bi, C.: 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, pp. 388–391. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/11428817_44

5. Derczynski, L., Maynard, D., Aswani, N., Bontcheva, K.: Microblog-genre noise and impact on semantic annotation accuracy. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media - HT '13. pp. 21–30. ACM Press, New York, New York, USA (May 2013)
6. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Communications of the ACM 51(12), 68–74 (2008)
7. Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., Dredze, M.: 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. pp. 80–88. CSLDAMT '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
8. Forsythand, E.N., Martell, C.H.: Lexical and Discourse Analysis of Online Chat Dialog. In: International Conference on Semantic Computing (ICSC 2007). pp. 19–26. IEEE (Sep 2007)
9. Foster, J., Çetinoglu, O., Wagner, J., Le Roux, J., Hogan, S., Nivre, J., Hogan, D., Van Genabith, J.: #hardtoparse: POS Tagging and Parsing the Twitterverse. In: AAAI 2011 Workshop On Analyzing Microtext. pp. 20–25 (2011), http://hal.archives-ouvertes.fr/hal-00702445
10. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: 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. pp. 42–47. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
11. Greene, B.B., Rubin, G.M.: Automatic Grammatical Tagging of English. Department of Linguistics, Brown University (1971)
12. Hachman, M.: Humanity's Tweets: Just 20 Terabytes — News & Opinion — PCMag.com (2011)
13. Leon Derczynski, Alan Ritter, S.C.: Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. Proceedings of the International Conference on Recent Advances in Natural Language Processing (2013)
14. Monojit Choudhury, Rahul Saraf, Vijit Jain, Sudeshna Sarkar, A.B., Choudhury, M., Saraf, R., Jain, V., Sarkar, S., Basu, A.: 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 (2007)
15. Nebhi, K.: Ontology-Based Information Extraction from Twitter. In: 24th International Conference on Computational Linguistics. p. 17 (2012)
16. Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. In: HLT-NAACL04. pp. 329–336 (2004)
17. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Natural language processing and text mining, pp. 9–28. Springer (2007)
18. Ritter, A., Cherry, C., Dolan, B.: Unsupervised Modeling of Twitter Conversations. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 172–180. HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
19. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: an experimental study pp. 1524–1534 (Jul 2011)
20. Roman Klinger, Katrin Tomanek, R.K.: Classical Probabilistic Models and Conditional Random Fields
21. Soderland, S., Cardie, C., Mooney, R.: Learning Information Extraction Rules for Semi-structured and Free Text. In: Machine Learning. pp. 233–272 (1999)
22. Sutton, C., McCallum, A.: An introduction to conditional random fields. arXiv preprint arXiv:1011.4088 (2010)