HMM POS Tagging of Twitter's Tweet

Muhammad Alif Akbar
School of Computing, Informatics Departement
Telkom University

***Abstract*—Twitter has become a main stream channel of communication. With a lot of conversation took place on it, Twitter can be a source to learn many things. The project build an HMM classifier to map a words in a sentence (tweet) on twitter into their proper tags. The HMM is build using unigram, bigram, and trigram model that calculate normalized word-tag. The method has proven effective with accuracy up to 94% for the task.**

***Index Terms*—Pos tagging, twitter, HMM, NLP.**

1. Introduction

Twitter has become a main stream channel of communication. With a lot of conversation took place on it, Twitter can be a source to learn many things. But, the conversation is just too big to be learned by a human so machine learning solution should be built. Natural Language Processing (NLP) has been an interesting research recently [1], [2]. One step on NLP is POS Tagging. Even the POS tagging is easy on well-structured sentences. It can be a challenge on unstructured sentence like be found on twitter. A sentence can consist of URL, Hashtag, Username, Symbol(emoticon) and Re-Tweet also slang words such as "LoL", "Hahaha", etc. So our project tried to do automatic tagging because with a good tagging every sentence can be learned further such as information retrieval (IR), sentiment analysis, etc.

2. Proposed Method

This project proposed to build classifier based on HMM to classify words into their tags. The project use tags that defined by Penn Tree Bank (PTB) that consist of 45 tags [3]. But those tags are not enough to map tweet sentences, so there are added 4 more tags that are URL, USR, HT, and RT. Then the data are passed into pre-processing, processing and testing.

2.1. Twitter

Twitter is one on notable website of social network. Twitter founded in 2006 by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams [4]. By 2013 Twitter generated 340 million tweet per day [5]. A sentence (tweet) in twitter can only consist of 140 characters that also containt url(regex: http://[.]+\.[.]+), hashtags (regex: #[.]+), symbols and username (regex: @[.]+).

2.2. Dataset

Dataset are obtained from GATE Twitter part-of-speech tagger [8]. It consist of 159492 sentences with words are already paired by their POS tag. The dataset then split by 70-30 into training and testing dataset, 111644 and 47848 respectivly. This done to simulate unknown data to the system. The tags that already on dataset is defined by Penn Tree Bank (PTB) P.O.S Tags and added 4 Tags to define twitter word class, hashtag, username, re-tweet, and url link with total 49 tags. The total appearance of tags is 1543040 as can be seen on table 1 and table 2.

2.3. HMM

Data processed using a Hidden Markov Model (HMM) classifier with the viterbi algorithm. HMM is chosen because it's simplicity and high accuracy for P.o.S Tagging [6], [7]. HMM is a classifier belong to statistical/bayesian family based on transition and emission probability. The HMM using will based on combination probability using unigram, bigram, and trigram word model. The HMM model is formulated to

$$P(w_1..n|tag_1..n) = \prod_{i=1}^{n} emission(w_i|t_i) \prod_{i=1}^{n} transition(t_i|t_{i-1})$$

Emision probability calculated using formula:

$$e = \frac{count(word_i, tag_i)}{count(tag_i)}$$

Transition probability calculated using formula based on unigram of the tags:

$$t = \frac{count(tag_i)}{\sum_i^n count(tag_i)}$$

Transition probability calculated using formula based on bigram of the tags:

$$t = \frac{count(tag_i, tag_{i-1})}{count(tag_{i-1})}$$

Transition probability calculated using formula based on trigram of the tags:

$$t = \frac{count(tag_i, tag_{i-1}, tag_{i-2})}{count(tag_{i-1}, tag_{i-2})}$$

TABLE 1: Tags Appearance on Training

Tag Definition	Tag Code	Appearance
Dollar	$	10
Opening quotation mark	“	642
Closing quotation mark	”	1162
Opening parenthesis	(	736
Closing parenthesis	)	1699
Comma	,	15897
Dash	–	0
Sentence terminator	.	72858
Colon or ellipsis	:	43282
Conjunction, coordinating	CC	15790
Numeral, cardinal	CD	8917
Determiner	DT	52546
Existential there	EX	270
Foreign word	FW	27
Twitter Hash Tag	HT	29030
Prep or conj, subor	IN	63300
Adjective or numeral, ordinal	JJ	44422
Adjective, comparative	JJR	1742
Adjective, superlative	JJS	2184
List item marker	LS	0
Modal auxiliary	MD	14952
Noun, singular or mass	NN	106256
Noun, proper, singular	NNP	31274
Noun, proper, plural	NNPS	66
Noun, common, plural	NNS	27861
Pre-determiner	PDT	3
Genitive marker	POS	1549
Pronoun, personal	PRP	102569
Pronoun, possessive	PRP$	20311
Adverb	RB	55515
Adverb, comparative	RBR	330
Adverb, superlative	RBS	193
Particle	RP	4449
Twitter re-tweet	RT	28327
Symbol	SYM	24
To	TO	20219
Interjection	UH	36624
Twitter URL	URL	12956
Twitter username	USR	71891
Verb, base form	VB	53364
Verb, past tense	VBD	19524
Verb, present participle	VBG	19864
Verb, past participle	VBN	7801
verb, present, singular	VBP	54885
verb, present, 3rd person	VBZ	19929
WH-determiner	WDT	51
WH-pronoun	WP	5658
WH-pronoun, possessive	WP$	0
Wh-adverb	WRB	8574

TABLE 2: Tags Appearance on Testing

Tag Definition	Tag Code	Appearance
Dollar	$	2
Opening quotation mark	“	255
Closing quotation mark	”	450
Opening parenthesis	(	291
Closing parenthesis	)	730
Comma	,	6660
Dash	–	0
Sentence terminator	.	31026
Colon or ellipsis	:	18576
Conjunction, coordinating	CC	6809
Numeral, cardinal	CD	3459
Determiner	DT	22515
Existential there	EX	107
Foreign word	FW	6
Twitter Hash Tag	HT	12245
Prep or conj, subor	IN	26943
Adjective or numeral, ordinal	JJ	18873
Adjective, comparative	JJR	666
Adjective, superlative	JJS	872
List item marker	LS	0
Modal auxiliary	MD	6661
Noun, singular or mass	NN	45864
Noun, proper, singular	NNP	13256
Noun, proper, plural	NNPS	30
Noun, common, plural	NNS	11734
Pre-determiner	PDT	0
Genitive marker	POS	631
Pronoun, personal	PRP	144906
Pronoun, possessive	PRP$	8834
Adverb	RB	24277
Adverb, comparative	RBR	134
Adverb, superlative	RBS	125
Particle	RP	1997
Twitter re-tweet	RT	12076
Symbol	SYM	5
To	TO	8703
Interjection	UH	15621
Twitter URL	URL	5283
Twitter username	USR	30858
Verb, base form	VB	23053
Verb, past tense	VBD	8354
Verb, present participle	VBG	8491
Verb, past participle	VBN	3288
verb, present, singular	VBP	24013
verb, present, 3rd person	VBZ	8518
WH-determiner	WDT	31
WH-pronoun	WP	2494
WH-pronoun, possessive	WP$	0
Wh-adverb	WRB	3687

as can be seen on table 1 and 2, there are tags of PTB that never appear neither on training sets nor testing sets. This will become a problem later on calculating joint probabilities.

3. Experiment and Testing

3.1. Preprocessing

Using HMM means the project need to conditional probability tables (CPTs). Which mean each word and tags are need to be counted and calculate their appearance probability. There is a problem in building cpt$s$ that is counting word. The problem in counting word is that some words are uncommon in the set (appear very little). This can be due to they are slang words, mistyped, shortened, person name, numerical, etc. This can lead to very big word set leading into very long algorithm execution time.

So to handle the problem the project normalize low-frequent words. In training process, words are counted as it is, then the table count is re-iterate to look for apperance count below 5 ($count(word_i) < 5$) to normalize them and set them a new word-class. In testing process, words are checked if exist on the cpt then get its probability or else if not exist the words are also normalized. The word-class are defined by Bicke et. (1999), can be seen on table 3:

TABLE 3: Low-Freq World Class

Word class	Intuition
twoDigitNum	two digit year
FourDigitNum	four digit year
ContainsDigitAndAlpha	product code
ContainsDigitAndDash	date, dash
ContainsDigitAndSlash	date, slash
ContainsDigitAndComma	monetary amount
ContainsDigitAndPeriod	monetary amount, percentage
OtherNum	other number
AllCaps	organization
CapPeriod	personal name initial
FirstWord	first word capital, sentence
InitCap	first word capital
Lowercase	uncapitalized
Other	other, symbol
OtherCap	other capital

3.2. Processing

The project build using HMM that put combination of unigram, bigram, and trigram word model into probability calculation. With each probabilty muliplied by a constant λ_i, with $\sum \lambda_i = 1$. Combination of unigram, bigram, and trigram hoped to increase the accuracy of system.

$$transition = \lambda_1 \times t_{unigram} + \lambda_2 \times t_{bigram} + \lambda_3 \times t_{trigram}$$

'Zero occurance' problem is another problem in building cpt. When an unknown word or tags put into the classifier system. 'Zero occurance' can lead into classifier error or zero result calculation. Which is bad because can reduce system accuracy. As been seen on the table 1 and 2 there are some tag that never been seen in training nor testing set. So to cancel zero occurance every tags appearance is added by 1 (+1) so that every tags are considered to appear at least 1. Also the project do not use any outside corpus so smoothing also done to every word-tag pair, (added by 1) which mean every words considered is paired with any tag at least 1 time. This smoothing lead a bit change to both emission and transision formul Emision probability calculated using formula:

$$e = \frac{count(word_i, tag_i) + 1}{count(tag_i) + \sum_{i=1}^{n} 1}$$

Transition probability calculated using formula for unigram matrix of the tags:

$$t_{unigram} = \frac{count(tag_i) + 1}{\sum_{i}^{n} count(tag_i) + \sum_{i=1}^{n} 1}$$

Transition probability calculated using formula for unigram matrix of the tags:

$$t_{bigram} = \frac{count(tag_i, tag_{i-1}) + 1}{count(tag_{i-1}) + \sum_{i=1}^{n^2} 1}$$

Transition probability calculated using formula for trigram matrix of the tags:

$$t_{trigram} = \frac{count(tag_i, tag_{i-1}, tag_{i-2}) + 1}{count(tag_{i-1}, tag_{i-2}) + \sum_{i=1}^{n^3} 1}$$

'zero occurance' must be eliminated because whenever a word or tag chain on testing set never appear on training corpus the probaility of whole sentence hosted the word will valued zero (0), and the tag chain will never be selected as possible outcome.

3.3. Testing and Result

HMM is pretty good classifier, but it needs to check all possible tags for a sentence $P(w_{1..n}|tag_{1..n}$ that needs a lot of execution time. So to make the algorithm efficent viterbi algorithm put to calculate the HMM. Viterbi is an algorithm belong to dynamic programming (DP) which build a very big matrix to speed up the looping process.

The constant used are $\lambda_1 = 0.1, \lambda_1 = 0.2, \lambda_1 = 0.7$ produce result with accuracy up to 94% for tag prediction and accuracy up to 63% for tweet prediction (whenever a tag is wrong in a tweet, the tweet count as false).

4. Conclusion

HMM with combination of unigram, bigram, trigram is proved as a very good classifier for Tweet POS tagging with accuracy up to 93%. The result could be observed more by changing the multiplier constant. A lower accuracy is observed whenever the constant of unigram model is much higher than the rest.

5. Future Work

The project only focused on tagging process without any information learned. An information retrieval can be build as a future work of the project to learn about what is going on in the tweet. Also an sentiment analysis can be build to measure about some topic populatrity, etc.

6. Acknowledgments

The authors would like to thank Said Al-Faraby as lecturer on NLP class and Coursera online course on NLP topic that help me to understand HMM. Also IEEE to give a latex format document that help me to write this report [9].

References

[1] D. Kumawat and V. Jain, POS Tagging Approaches: A Comparison, IJCAI , vol. 118, no. 6, pp. 3238, May 2015.

[2] S. Yin and G. Fan, Research of POS Tagging Rules Mining Algorithm, in Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), China, 2013.

[3] E. Atwell, *The University of Pennsylvania (Penn) Treebank Tag-set*. [Online]. Available: http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html. [Accessed: 16-Dec-2016].

[4] Twitter. [Online]. Available: https://en.wikipedia.org/wiki/Twitter. [Accessed: 16-Dec-2016]

[5] Twitter (March 21, 2012). *Twitter turns six*. Twitter.

[6] F. M. Hasan, N. UzZaman, and M. Khan, Comparison of different POS Tagging Techniques (N-Gram, HMM and Brills tagger) for Bangla, in Advances and Innovations in Systems, Computing Sciences and Software Engineering, Springer, 2007, pp. 121126.

[7] J. Van Gael, A. Vlachos, and Z. Ghahramani, The Infinite HMM for Unsupervised PoS Tagging, in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, Singapore, 2009, pp. 678687.

[8] L. Derczynski, A. Ritter, S. Clarke, and K. Bontcheva. 2013. *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, ACL.

[9] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.