Named Entity Recognition for Nepali Text Using Support Vector Machines

Article *in* Intelligent Information Management · January 2014

DOI: 10.4236/iim.2014.62004

Tej Bahadur Shahi
Tribhuvan University

Intelligent Information Management, 2014, 6, 21-29
Published Online March 2014 in SciRes. http://www.scirp.org/journal/iim
http://dx.doi.org/10.4236/iim.2014.62004

Named Entity Recognition for Nepali Text Using Support Vector Machines

Surya Bahadur Bam, Tej Bahadur Shahi

Central Department of Computer Science and Information Technology (CDCSIT), Tribhuvan University, Kirtipur, Nepal
Email: tejshahi1984@yahoo.com

Received 13 December 2013; revised 12 January 2014; accepted 10 February 2014

Abstract

Named Entity Recognition aims to identify and to classify rigid designators in text such as proper names, biological species, and temporal expressions into some predefined categories. There has been growing interest in this field of research since the early 1990s. Named Entity Recognition has a vital role in different fields of natural language processing such as Machine Translation, Information Extraction, Question Answering System and various other fields. In this paper, Named Entity Recognition for Nepali text, based on the Support Vector Machine (SVM) is presented which is one of machine learning approaches for the classification task. A set of features are extracted from training data set. Accuracy and efficiency of SVM classifier are analyzed in three different sizes of training data set. Recognition systems are tested with ten datasets for Nepali text. The strength of this work is the efficient feature extraction and the comprehensive recognition techniques. The Support Vector Machine based Named Entity Recognition is limited to use a certain set of features and it uses a small dictionary which affects its performance. The learning performance of recognition system is observed. It is found that system can learn well from the small set of training data and increase the rate of learning on the increment of training size.

Keywords

Support Vector Machine; Named Entity Recognition; Machine Learning; Classification; Nepali Language Text

1. Introduction

The term Named Entity (NE) was evolved during the sixth Message Understanding Conference (MUC-6, 1995) [1]. NE is the structured information referring to predefined proper names, like persons, locations, and organiza-

How to cite this paper: Bam, S.B. and Shahi, T.B. (2014) Named Entity Recognition for Nepali Text Using Support Vector Machines. *Intelligent Information Management*, **6**, 21-29. http://dx.doi.org/10.4236/iim.2014.62004

tions etc. NE task is to identify all named locations, named persons, named organizations, date, times, monetary amounts, percentages etc. in text.

Named Entity Recognition (NER) aims to classify each word of a document into predefined target named entity classes and is nowadays considered to be fundamental activity for many Natural Language Processing (NLP) tasks such as information retrieval, machine translation, information extraction, question answering systems [1] [2]. Though Support Vector Machine (SVM) [3] technique has been widely applied to NER in several well-studied languages, the use of SVM technique to Nepali Languages (NLs) is very new. The basic principle of proposed Named Entity Recognition is illustrated in **Figure 1**.

1.1. Challenges of NE Recognition for Nepali Text

NE recognition in Nepali languages is difficult and challenging as:

No Capitalization

English and many other European languages use capitalization to recognize proper names. Orthography of Nepali does not support capitalization.

Agglutinative Nature

Agglutinative means that some additional features can be added to the word to add more complex meaning. Agglutinative language form sentences by adding a suffix to the root forms of the word. Nepali is a highly inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex word forms. For example, let us consider the root word as राजा (**king**) and suffix as ईश्वर (**God**) then if we combine these two words then it becomes राजेश्वर (**a name of person**) as new word.

Proper Name Ambiguity

Ambiguity in proper name present in Nepali language as in English. The names like White are ambiguous in English-name or color. Nepali Person Names are more diverse compared to the other languages and a lot of these words can be found in the dictionary with some other specific meanings. There is a surprising amount of ambiguity even among proper names. For example:

- People vs. Companies: टाटा, फोर्ड etc.
- People vs. Locations: पशुपति (Name of Temple).
- People vs. Organizations: त्रिभुवन (person vs. university).
- Acronyms vs. Organizations: MRI (Magnetic Resonance Imaging vs. Mental Research Institute).
- People vs. Months: बैशाख (month of Nepali Calendar).

Word Order

Languages like Nepali have a different word-order than English and some have a free word-order. Nepali mostly has a word order but depending upon the domain the word order is respected. For example, *कमलले पानीको पूरा गिलास पियो र पानीको गिलास कमलले पुरा पियो both translate to* ***Kamal drank a whole glass of water***.

Loan Words in Nepali

Nepali has a number of loan words. Loan words are words that are not indigenous to Nepali. The named entity recognizer that is based on simple morphological cues will fail to recognize a large number of proper nouns. For example Osama Bin Laden, बिन (Bin) an Arabic cue needs to be used in the middle of the name for the per-

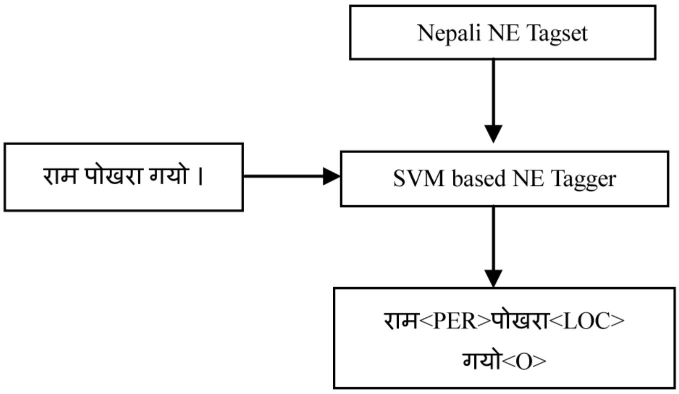

Figure 1. Example of Named Entity Tagger.

son name.

Nested Entities

The named entities that are classified as nested contain two proper names that are nested together to form a new named entity. An example in Nepali is Kathmandu University where Kathmandu is the location name and University marks the whole entity as an organization.

Resource Challenges

NER approaches are either based on rule engine or inference engines. In each approach some type of corpus is required; lack of a NE tagged corpus for deriving rules is an issue for Nepali language. Nepali is a resource poor language annotated corpora, name dictionaries; good morphological analyzers, POS taggers etc. are not yet available in the required measure. Although Nepali language have a very old and rich literary history, technological development are of recent origin. Web sources for name lists are available in English, but such lists are not available in Nepali forcing the use of transliteration for creating, such lists.

1.2. Support Vector Machine

In their basic form shown in **Figure 2**, SVM construct the hyperplane in input space that correctly separate the example data into two classes. Hence SVM is a binary classifier. This hyperplane can be used to make the prediction of class for unseen data. The hyperplane always exist for the linearly separable data [4].

2. Related Work

Considerable amount of work has already been done in the field of NER for English and other language like German, Spanish, Chinese, and Bengali etc. But there is no any work for Nepali language has been done yet. Different approaches like the rule based approach, the stochastic approach and the transformation based learning approach along with modification have been tried and implemented for English and European language. However, if we look at the same scenario for South-Asian language such as Bangla, Hindi, and Nepali, we find out that not much work has been done in the area of NER.

The author of [3] [5] had shown that Conditional Random Fields (CRFs) are undirected graphical models used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes. A conditional random field (CRF) is a type of discriminative probabilistic model used for the labeling sequential data such as natural language text. The author of [6] had shown that the maximum entropy [ME] [3], framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived from training data, expressing some relationship between features and outcome. The author of [7] had shown that Name recognition may be viewed as a classification problem, where every word is either part of some name or not part of any name. In recent years, hidden Markov models (HMM's) have enjoyed great success in other textual classification problems most notably part-of-speech tagging [8]. The decision tree [9] uses part of speech, character type, and special diction-

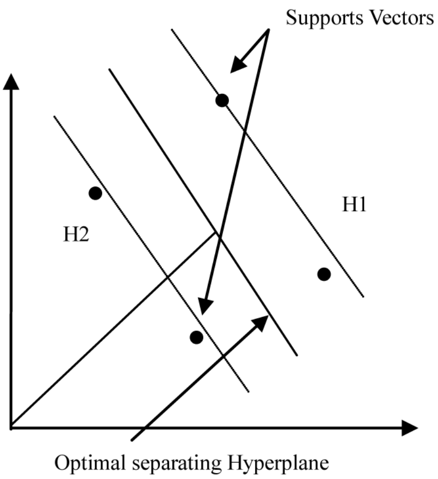

Figure 2. Support Vector Machine.

nary information to determine the probability that a particular type of name opens or closes at a given position in the text. Support Vector Machines (SVMs) based NER system was proposed in [10] for Japanese. His system is an extension of Kudo's chunking system [11] that gave the best performance at CoNLL-2000 shared tasks. The other SVM-based NER systems can be found in [2] [12].

3. Methodology: A Support Vector Machine Approach

The proposed model in this work for NER is given in **Figure 3**. This describes top level data flow diagram of NER problem, used in this work. The proposed system framework consist preprocessing, feature extraction, training and learning the data for support vector machine algorithm.

Preprocessing engine has Nepali corpus as input. After preprocessing, only important data is stored and feature vectors are extracted from the preprocessed data. Training corpus is the most powerful and is the heart of Named Entity Recognition

3.1. Data Preparation

Training data must be in column format, *i.e*. a word per line corpus in a sentence by sentence fashion. The column separator is the only one blank space. The word is expected to be the first column of the line. The tag to predict takes the second column in the output. Following is a sample of the training data:

राम <PER>

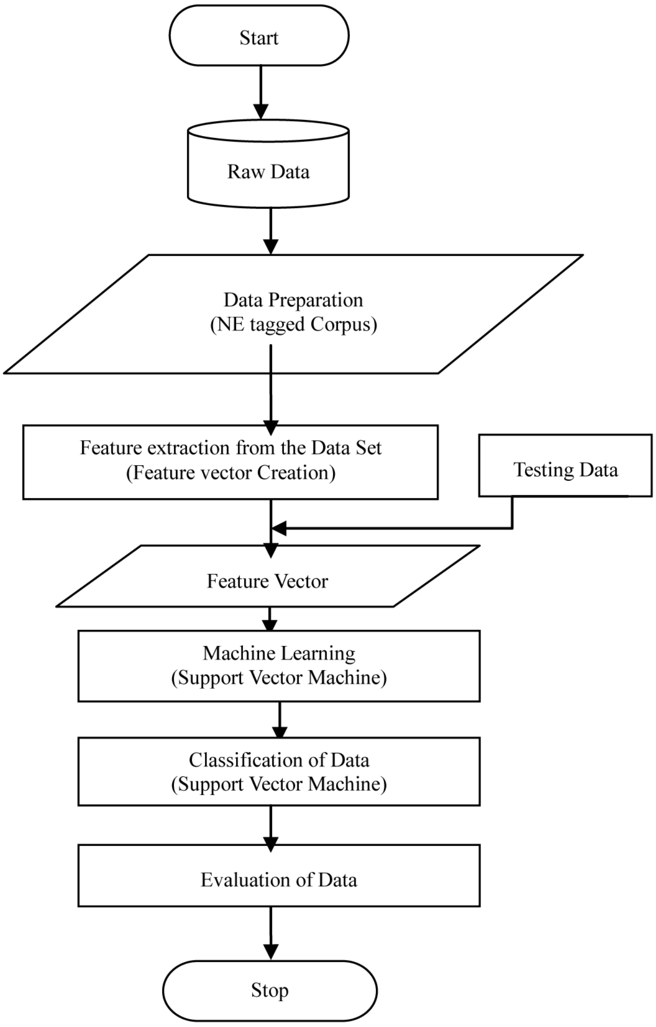

Figure 3. Implementation Model for Nepali NER.

पोखरा <LOC>
गयो <O>
………..
………...

3.2. Feature Extraction

The features used in this work are taken form [13]. Following are the details of the set of features that will be apply to solve the NER task for Nepali Text:

- First word: This is used to check whether the current token is the first word of the sentence or not. Though Nepali is relatively free order languages, the first word of the sentence is most likely a NE as it appears in the subject position most of the time.
- Word length: This binary valued feature is used to check whether the length of the current word is less than two or not. This is based on the observation that the very short words are rarely NEs.
- Digit features: Several binary valued digit features have been defined, depending upon the presence and/or the number of digits in a token (e.g., ContainsDigit [token contains digits], FourDigit [token consists of four digits], TwoDigit [token consists of two digits]), combination of digits and punctuation symbols (e.g., ContainsDigitAndComma [token consists of digits and comma], combination of digits and symbols (e.g., ContainsDigitAndSlash [token consists of digit and slash], ContainsDigitAndHyphen [token consists of digits and hyphen], ContainsDigitAndPercentage [token consists of digits and percentages]). These binary valued features are helpful in recognizing miscellaneous NEs, such as time expressions (Age, Date, Year), measurement expressions (Weight, Height etc) and numerical numbers etc.
- Gazetteer Lists: Various gazetteer lists are used.
    - Person name: This list contains the name of persons. The feature PersonName is set to +1 for the current word.
    - Location name: This list contains the location names and the feature LocationName is set to +1 for the current word.
    - Organization name: This list contains the organization names and the feature OrgnizationName is set to +1 for the current word.
    - Month name: This list contains the name of all twelve different months of both English and Nepali calendars. The feature MonthName is set to +1 for the current word.
    - Day name: This list contains the name of all seven different days of Nepali calendars. The feature DayName is set to +1 for the current word.
    - PersonPrefix: This list contains the person prefix such as श्री, श्रीमान, श्रीमति etc.
    - MiddleName: This list contains nepali middle name such as बहादुर, कुमार, कुमारी, देबी, राज, प्रसाद etc.
    - SurName: This list contains nepali sur name such as बम, पन्त, जोशि, भट्ट, दाहाल etc.
    - CommonLocationWord: This list contains common location word such as रोड, बाटो, राजमार्ग, नगर etc.
    - Action Verb: A set of action verbs like सुन, भन, गर, खाउ, जाउ etc. often determine the presence of person names. Person names generally appear before action verbs.
    - Designation Word: This list contains designation word such as प्रोफेशर, डा., मन्त्रि, रास्ट्रपति, सचिब, अध्यक्ष, महासचिव etc.
    - Organization Suffix Word: This list contains organization suffix word such as मिल, प्रालि, कम्पनि, समिति, संघ, कार्यालय etc.

3.3. One vs Rest Classification

The SVM described in section 1.2 is used for binary classification and which classify data in binary class. But Named entity recognition is a multiclass classification problem since in natural language there are more than two NE tags. As an instance, for this work, the five tags as listed in **Table 1** are used to cover all grammatical categories and in which four tags are NE and fifth tag is used to represent the word which does not belongs to the named entity *i.e*. other than NE. In this work number of tag represents the number of classes. So binarization of problem must be performed before apply them to NE tagging. [14] has suggested the one vs. rest binarization of problem *i.e*. a SVM is trained for each NE tag in order to distinguish this class and the rest. When tagging the word, the most confident prediction among the all binary SVM is selected.

But in the case of NER there are five classes, so multiclass SVM is used. Here five SVM are trained that corresponds to five NE tag and for each new word, each of five SVM are evaluated and most confident NE tag is assigned to that word. This can be explained with an example as in Figure 4.

3.4. Tool Used

For this work, the SVM multiclass [15] is used. SVM multiclass is an implementation of Support Vector Machines (SVMs) in C programming language. Main feature of this system is that we can integrate our own custom kernel very easily. Because of steepest feasible descent and caching of kernel evaluations, SVM multiclass is real fast. It can easily handle thousands of support vectors and several hundred-thousands of training examples. At first, system learns from training file using customized kernel function and creates a model file. Model file basically learn all the support vectors. This model file is used for classifying new examples. After testing is complete, it produces a prediction file which contains the confidence value of each example for that classification.

4. Result and Discussion

The study has gone through the empirical analysis of the performance of the NE recognizer. Here, during the development of the model, the impact of the size of the training data and test data on the performance was observed. The experiment was done for three different sizes of the train data; it is shown that the performance of the method depends on the size of train data.

In the Experiment No 1 shown in Table 2, the training data set consists of 5000 tokens and the SVM is trained with these tokens and tested with 10 different test data sets from size 1000 tokens to 5500 tokens.

In the Experiment No 2 shown in Table 3, the training data set consists of 15,000 tokens and the SVM is trained with these tokens and tested with 10 different test data sets from size 1000 tokens to 5500 tokens.

In the Experiment No 3 shown in Table 4, the training data set consists of 29,298 tokens and the SVM is trained with these tokens and tested with 10 different test data sets from size 1000 tokens to 5500 tokens.

The Table 5 shows the average result of each of above three experiments. The results for experiment no. 1 is 65.93% precision, 80.42% recall, and 72.44% F-score, (taken as average of ten runs), for Experiment No. 2 is

Table 1. Named Entity Tag-set for Nepali NER.

NE Tag	Meaning	NE examples
PER	Person name	जनक <PER> जनकजोशी<PER>
LOC	Location name	बुटवल<LOC> बुटवलराजमार्ग<LOC>
ORG	Organization name	त्रिभुवनविश्वविध्यालय<ORG>
MISC	Miscellaneous name	बैशाख<MISC>
O	Words that are not NE	भिर <O>, पेश <O>

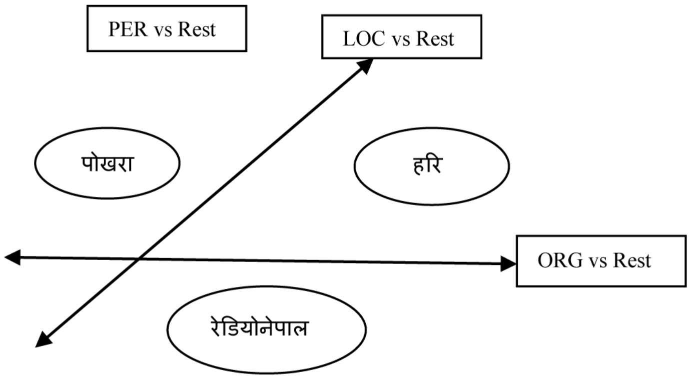

Figure 4. Multi Class SVM Example.

Table 2. Experiment No. 1 (Training Size 5000 tokens).

Exp.No.	Size of Test Data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	69.61	80.46	74.64
2	1500	68.08	80.41	73.73
3	2000	67.88	80.53	73.67
4	2500	62.07	80.58	70.13
5	3000	63.40	80.52	70.94
6	3500	64.69	80.26	71.64
7	4000	67.02	80.50	73.14
8	4500	65.48	80.42	72.19
9	5000	65.06	80.38	71.91
10	5500	66.02	80.22	72.43

Table 3. Experiment No. 2 (Training Size 15,000 tokens).

Experiment No.	Size of Test data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	82.58	97.96	89.62
2	1500	84.84	97.49	90.73
3	2000	86.57	98.51	92.15
4	2500	79.52	94.72	86.46
5	3000	75.53	95.73	84.44
6	3500	83.12	98.08	89.98
7	4000	84.42	97.91	90.66
8	4500	81.41	97.12	88.57
9	5000	84.63	98.16	90.90
10	5500	84.07	97.09	90.11

Table 4. Experiment No. 3 (Training Size 29,298 tokens).

Experiment No.	Size of Test data (in tokens)	Precision (%)	Recall (%)	F-Score (%)
1	1000	89.51	98.99	94.01
2	1500	88.96	98.72	93.59
3	2000	90.76	99.29	94.83
4	2500	81.51	97.61	88.84
5	3000	86.15	98.56	91.94
6	3500	85.57	98.57	91.61
7	4000	88.57	98.77	93.39
8	4500	86.47	98.41	92.06
9	5000	85.85	98.66	91.81
10	5500	85.19	97.80	91.06

Table 5. The Precision, Recall and F-Score for different training data size.

Training Data Size (in tokens)	Precision	Recall	F-Score
5000	65.93%	80.42%	72.44%
15,000	82.66%	97.27%	89.36%
29,298	86.85%	98.53%	92.31%

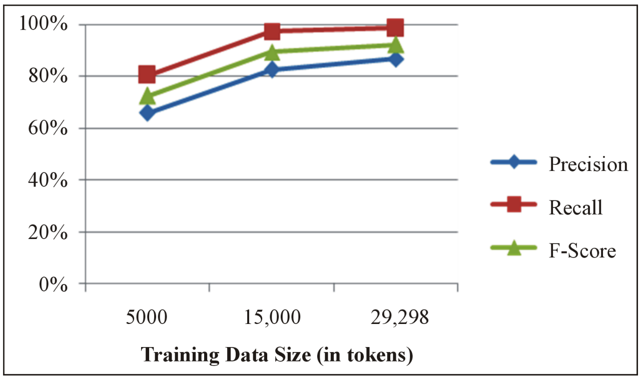

Figure 5. Learning curve for SVM based NE tagger.

82.66% precision, 97.27% recall, and 89.36% F-score, (taken as average of ten runs), The result for experiment no. 3 is 86.85%, precision, 98.53%, recall, and 92.31% F-score, (taken as average of ten runs). From these experiments, it is observed that the learning ability of SVM for NE recognition is increased when the size of training data is increased.

The learning curve corresponding to the result in **Table 5** is presented in **Figure 5**.

5. Conclusions and Future Work

In this work, the method for extracting named entities from data of various domains has been presented which is a system useful in the identification and classification of names. The work for Nepali NER is very complex due to the nature of Nepali language which is in free order and lacks of research work in Nepali text. There is no any corpus existing for Named Entity so it is difficult and tedious to create such corpus. For this work, the NE corpus is created manually.

The corpus used is comparatively small with respect to other languages and its size can be increased in future. Other classification methods may also be tested for the recognition of NER in future.

Acknowledgements

Authors would like to thank Prof. Dr. Shashidhhar Ram Joshi and Prof. Dr. Tank Nath Dhamala for their supervision during the completion of this work.

References

[1] Bindu, M.S and Idicula, S.M. (2011) Named Entity Recognizer employing Multiclass Support Vector Machines for the Development of Question Answering Systems. *International Journal of Computer Applications* (0975-8887), **25**.

[2] Asif, E. and Sivaji, B. (2008) Bengali Named Entity Recognition Using Support Vector Machine. *Proceedings of the IJCNLP*08 *Workshop on NER for South and South East Asian Languages*, Hyderabad, 12 January 2008, 51-58.

[3] Wu, Y.C., Fan, T.K., Lee, Y-S. and Yen, S.-J. (2006) Extracting Named Entities Using Support Vector Machines. Springer- Verlag, Berlin.

[4] Asif, E. and Sivaji, B. (2010) Named Entity Recognition Using Appropriate Unlabeled Data. *Post-Processing and Voting Informatica*, **34**, 55-76.

[5] Sobhan, N.V., Pabitra, M. and Ghosh, S.K. (2010) Conditional Random Field Based Named Entity Recognition in

Geological Text Sobhana. *International Journal of Computer Applications* (0975 - 8887), **1**.

[6] Joel, N. (2008) Learning NER from Wikipedia.

[7] Bikel, D.M., Schwartz, R.L. and Weischedel, R.M. (1999) An Algorithm that Learns What's in a Name. *Machine Learning*, **34**, 211-231. http://dx.doi.org/10.1023/A:1007558221122

[8] Zhou, G. and Su, J. (2002) Named Entity Recognition Using an HMM-Based Chunk Tagger. *Proceedings of the* 40*th Annual Meeting of the Association for Computational Linguistics* (*ACL*'2002), Philadelphia, July 2002, 473-480.

[9] Antonio, T., Rafael, M. and Monica, M. (2008) Named Entity WordNet. *In Proceedings of the 6th International Language Resources and Evaluation Conference*, 2008.

[10] Yamada, H., Kudo, T. and Matsumoto, Y. (2001) Japanese Named Entity Extraction Using Support Vector Machine. *Transactions of IPSJ*, **43**, 44-53.

[11] Kudo, T. and Matsumoto, Y. (2001) Chunking with Support Vector Machines. *Proceedings of NAACL*, **200**, 192-199

[12] Asif, E. and Sivaji, B. (2010) Named Entity Recognition Using Support Vector Machine: A Language Independent Approach. *International Journal of Electrical and Electronics Engineering*, **4**, 155.

[13] Bam, S., (2013) Support Vector Machine Based Named Entity Recognition for Nepali Text. Masters Dissertation, Central Department of Computer Science and IT, Tribhuvan University, Kirtipur.

[14] Shahi, T.B., (2012) Support Vector Machine Based POS Tagging for Nepali Text. Masters Dissertation, Central Department of Computer Science and IT, Tribhuvan University, Kirtipur.

[15] Joachims, T. (2008) Multi-Class Support Vector Machine. Cornell University, Ithaca.