Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Comprehensive Study of Named Entity Recognition on Inflectional Languages

Arindam Dey, Md Jaynal Abedin , Dr.Bipul Syam Purkayastha

Abstract: ***Named entity recognition (NER) is one of the fundamental task in Natural Language Processing. In medical domain, there have been a number of studies on NER in English clinical notes; very limited research has been carried out on inflectional languages. The goal of this study was to semantically investigate features and machine learning algorithms for NER in Inflectional Language. About 1000 sentences are collected randomly from different domains of an inflectional language. One third of 1000 sentence were used to train the NER systems and one third for testing. We investigated the effects of different types of feature including bag-of-characters, word segmentation, part-of-speech, and section information, and different machine learning algorithms including conditional random fields (CRF), support vector machines (SVM), maximum entropy (ME), and structural SVM (SSVM) on the Inflectional language NER task. All classifiers were trained on the training dataset and evaluated on the test set, and micro-averaged precision, recall, and F-measure were reported.***

Key Words: Named Entity Recognition, Natural Language processing, Conditional Random Field, Support vector Machine, Maximum Entropy.

I. Introduction

Named Entity Recognition is a task to discover the Named Entities (NEs) in a document and then categorize these NEs into diverse Named Entity classes.

The term "Named Entity", now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6)[9].Broadly speaking, named entities are proper nouns. However, named entity tasks often include expressions for date and time, names of sports and adventure activities, terms for biological species and substances as named entities. MUC- 7 classifies named entities into following categories and subcategories:

a. Entity (ENAMEX): person, organization, location
b. Time expression (TIMEX): date, time
c. Numeric expression (NUMEX): money, percent [5]

A. Named Entity Recognition and Classification (NER)

It was noticed that it is essential to recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions for various Information Extraction and NLP tasks. Identifying references to these entities in text was recognized as one of the important sub-tasks of IE and was called "Named Entity Recognition and Classification (NER)".

Though this sounds clear, special cases arise to require lengthy guidelines, *e.g*., when is *The Times of India* an artifact, and when is it an organization? When is *White House* an organization, and when a location? Are branch offices of a bank an organization? Is a garment factory a location or an organization? Is a street name a location? Is a phone number a numeric expression or is it an address (location). Is *mid-morning* a time? In order to achieve human annotator consistency, guidelines with numerous special cases have been defined for the Seventh Message Understanding Conference, MUC-7 .

Most research on NER systems has been structured as taking an un-annotated block of text, for *e.g*.: <PERSON>श्री हान</PERSON> फाइदा को श्रेय <ORG>कम्पनी</ORG> लाई <QUANTIFIER >सब </QUANTIFIER> भन्दा बढी जानकारी भएको कुरा मा ध्यान केन्द्रित गर्ने दर्शन लाई दिन्छन् ।[2]

B. Applications of NER

NER finds application in most of the NLP applications. The following list mentions few of its applications.

1) NER is very useful for search engines. NER helps in structuring textual information, and structured information helps in efficient indexing and retrieval of documents for search.

2) In the context of Cross-Lingual Information Access Retrieval (CLIR), given a query word, it is very important to find if it is a named entity or not. If a query word is a Named Entity, we need to transliterate a query word, rather than translating it.

3) The new generation of news aggregation platforms is powered by named entity recognition. A lot of information can be analysed using named entities, like plotting the popularity of entities over time and generating geospatial heat maps However, the main improvement to traditional news aggregation brought by NEs is how they connect between people and things.

4) NER finds application in machine translation, as well. Usually, entities identified as Named Entities are transliterated as opposed to getting translated.

5) Before reading an article, if the reader could be shown the named entities, the user would be able to get a fair idea about the contents of the article.

6) Automatic indexing of Books: Most of the words indexed in the back index of a book are Named Entities.

7) Useful in Biomedical domain to identify Proteins, medicines, diseases, *etc*.

8) NE Tagger is usually a sub-task in most of the information extraction tasks because it adds structure to raw information. [8]

II. APPROACHES OF NERS

There are three approaches of NERs. They are (i) Rule based approach and (ii) Statistical Approach and (iii) Hybrid Approach. [2][3][4][5]

The Rule Based Approach can either be List lookup Approach or a Linguistic Approach.

For NER detection using lookup approach or linguistic approaches, a lot of human effort is required. A large Gazetteer list has to be built for different Named Entity classes under lookup approach. Then, search operations are performed to find that the given word in the corpus is under which category of the Named Entity Classes. In a linguistic approach, a linguist set the rules and algorithms to determine NEs in a corpus and also classifies these NEs into respective Named Entity Classes.[1][6][7][8]

In Statistical Approach very less amount of human labour is required. It is an automated approach. It is of following types:

A. Hidden Markov Model(HMM)
B. Maximum Entropy Model(MEM)
C. Conditional Random Field(CRF)
D. Support Vector Machine(SVM)
E. Decision Tree(DT)[1][2]

In Hybrid Approach two approaches can be merged together. It improves the performance of NER system. It can be the combination of Linguistic and Statistical models like Gazetteer list and HMM, HMM and CRF or CRF and MEM etc.

A. HIDDEN MARKOV MODEL

When the state of a process cannot be inspected directly, it must be estimated from some sequence of observations. For example, the emotional state of another agent cannot be inspected without peeking into its head, but the emotional state is responsible for the agent's actions—so we should be able to estimate the agent's inner state by observing what it is doing.

Hidden Markov models are used to represent processes that are not fully observable. They augment the n-gram model with a set of actions that can be observed, and a probabilistic mapping between actions and states.

A first-order HMM is a tuple $M = \langle S, A, p, q \rangle$ where:

• S is the set of states in the process,

• A is the set of actions that can be observed,

• p is the transition probability function, where $p(s_t|s_{t-1})$ signifies the probability of transition from state s_{t-1} to state s_t, and

• q is the action observation probability function, where $q(a_t|s_t)$ denotes the probability of observing action a_t at time t given state s_t .

B. MAXIMUM ENTROPY MODEL

The Maximum Entropy model produces a probability distribution for the PP-attachment decision using only information from the verb phrase in which the attachment occurs. We denote the partially parsed verb phrase, i.e., the verb phrase without the attachment decision, as a history h, and the conditional probability of an attachment as p(d|h), where d€{0, 1} and corresponds to a noun or verb attachment (respectively). The probability model depends on certain features of the whole event (h, d) denoted by $f_i(h, d)$. An example of a binary-valued feature function is the indicator function that a particular (V, P) bigram occurred along with the attachment decision being V, i.e. $f_{print, on}(h, d)$ is one if and only if the main verb of h is "print", the preposition is "on", and d is "V". The ME principle leads to a model for p (d|h) which maximizes the training data log-likelihood,

$$\frac{\Sigma}{h,d}\bar{p}(h,d)logp(d|h)$$

where p˜(h, w) is the empirical distribution of the training set, and where p(d|h) itself is an exponential model:

$$\mathrm{p(d|h)} = \frac{\prod_{i=0}^{k} e^{\lambda_i f_i(h,d)}}{\sum_{d=0}^{1}\prod_{i=0}^{k} c^{\lambda_i f_i(h,d)}}$$

At the maximum of the training data log-likelihood, the model has the property that its k parameters, namely the λi's, satisfy k constraints on the expected values of feature functions, where the i^{th} constraint is,

$$E_m f_i = \bar{E} f_i$$

The model expected value is,

$$E_m f_i = \sum_{h,d} \bar{p}(h)p(d|h)f_i(h,d)$$

and the training data expected value, also called the desired value, is

$$\bar{E} f_i = \sum_{h,d} \bar{p}(h)p(d|h)f_i(h,d)$$

The values of these k parameters can be obtained by one of many iterative algorithms. For example, one can use the Generalized Iterative Scaling algorithm of Darroch and Ratcliff. As one increases the number of features, the achievable maximum of the training data likelihood increases.

C. CONDITIONAL RANDOM FIELD

CRFs are discriminative models, as they model the conditional distribution over labelling given some contextual observations, p(s|o), where s is the labelling and o is the context. This contrasts with generative models, which model the joint distribution over labelling and the context, p(s,o). These models are commonly used for decoding test instances where only the context is observed. In this case the maximising labelling of the conditional p(s|o) is required, $s^* = \mathrm{argmax}_s\ p(s|o)$. Discriminative models can be used directly in this instance, where generative models first require normalisation, $p(s|o) = \frac{p(s,o)}{\sum s' p(s',o)}$. This is an advantage of discriminative models, which are trained to maximise the conditional likelihood of the training sample. Discriminative models allow a richer feature representation, which provides more natural and accurate modelling. This benefit often
comes at the cost of increased training complexity and reduced flexibility with partially observed data. However, for many NLP tasks the advantages of discriminative models outweigh the disadvantages.

CRFs are most commonly used to model sequencing tasks, where the contextual observations are a sequence of tokens, **o** = $o_1, o_2, \ldots, o_N$, and the labelling is a sequence of labels of the same length, **s** = $s_1, s_2, \ldots, s_N$. This corresponds to labelling each token with a single label, as is the case for most tagging tasks. These sequencing CRFs are often referred to as linear chain CRFs; this refers to the chain graphical structure used to describe Markov assumptions over the label sequence. The name Conditional Random Field denotes the modelling of the labelling, S = **s**, as a network of inter-dependent random variables (a random field), while conditioning over another set of random variables: the context, O = **o**.

D. SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) algorithm (Cortes and Vapnik, 1995) is probably the most widely used kernel learning algorithm. It achieves relatively robust pattern recognition performance using well established concepts in optimization theory. Despite this mathematical classicism, the implementation of efficient SVM solvers has diverged from the classical methods of numerical optimization. This divergence is common to virtually all learning algorithms. The numerical optimization literature focuses on the asymptotical performance: how quickly the accuracy of the solution increases with computing time. In the case of learning algorithms, two other factors mitigate the impact of optimization accuracy.

Consider logistic regression, where the probability p(y = 1|x; θ) is modelled by is modelled by $h_\theta(x) = g(\theta^T x)$. We would then predict "1" on an input x if and only if $h_\theta(x) \geq 0.5$, or equivalently, if and only if $\theta^T x \geq 0$. Consider a positive training example (y = 1). The larger $\theta^T x$ is, the larger also is $h_\theta(x) = p(y = 1|x;w, b)$, and thus also the higher our degree of "confidence" that the label is 1. Thus, informally we can think of our prediction as being a very confident one that y = 1 if $\theta^T x \gg 0$. Similarly, we think of logistic regression as making a very confident prediction of y = 0, if $\theta^T x \ll 0$. Given a training set, again informally it seems that we'd have found a good fit to the training data if we can find θ so that $\theta^T x^{(i)} \gg 0$ whenever $y^{(i)} = 1$, and $\theta^T x^{(i)} \ll 0$ whenever $y^{(i)} = 0$, since this would reflect a very confident (and correct) set of classifications for all the training examples. This seems to be
a nice goal to aim for, and we'll soon formalize this idea using the notion of functional margins.

For a different type of intuition, consider the following figure, in which x's represent positive training examples, o's denote negative training examples, a decision boundary (this is the line given by the equation $\theta^T x = 0$, and is also called the separating hyper plane) is also shown, and three points have also been labelled A, B and C.

E. DECISION TREE

A likelihood-based approach to decision tree induction requires a probabilistic model of the process by which data are generated. For a given input x, we assume that a sequence of probabilistic decisions are taken that result in the generation

of a corresponding output y. We do not require that
this sequence of decisions have a direct correspondence to a
process in reality, rather the decisions may simply represent an abstract set of "twenty questions" that specify, with increasing precision, the location of the conditional mean of y on a nonlinear manifold that relates inputs to mean outputs.
We consider regression models in which yis a real-valued vector and classification models in which is either a binary scalar or a binary vector with a single non-zero component. In either case the goal is to formulate a conditional probability density of the formP(yjx;_), where _is a parameter vector. Maximizing a product of Nsuch densities with respect to _
(where Nis the sample size) yields a maximum likelihood estimate of _. Bayesian maximum a posterior estimation can be handled by incorporating a prior on the parameter vector. In a later section, we consider a Markov model in which the likelihood of a data sequence is not simply the product of N independent densities.

III. CURRENT STATUS IN NER FOR INDIAN LANGUAGES(ILS)

Although a lot of work has been done in English and other foreign languages like Spanish, Chinese etc. with high accuracy but regarding research in Indian languages is at initial stage only. Accurate NER systems are now available for European Languages especially for English and for East Asian language. For south and South East Asian languages the problem of NER is still far from being solved. There are many issues which make the nature of the problem different for Indian languages.
For example:- The number of frequently used words (common nouns) which can also be used as names (Proper nouns) is very large for European language where a large proportion of the first names are not used as common words.

IV. CHALLENGES IN NER

Named Entity Recognition was first introduced as part of Message Understanding Conference (MUC-6) in 1995 and a related conference MET-1 in 1996 introduced named entity recognition in non-English text. In spite of the recognized importance of names in applications, most text processing applications such as search systems, spelling checkers, and document management systems, do not treat proper names correctly. This suggests proper names are difficult to identify and interpret in unstructured text. Generally, names can have innumerable structure in and across languages. Names can overlap with other names and other words. Simple clues like capitalization can be misleading for English and mostly not present in non-western languages like Nepali.
The goal of NER is first to recognize the potential named entities and then resolve the ambiguity in the name. There are two types of ambiguities in names, structural ambiguity and semantic ambiguity. Wacholder et al. (1997) describes these ambiguities in detail. Non- English names pose another dimension of problems in NER e.g. the most common first name in the world is Muhammad, which can be transliterated as Mohmmed, Muhammad, Mohammad, Mohamed, Mohd and many other variations. These variations make it difficult to find the intended named entity. This transliteration problem can be solved if the name Muhammad is written in Arabic script as .محمد

V. RELATED WORKS

Although over the years there has been considerable work done for NER in English and other European languages, the interest in the South Asian languages has been quite low until recently. One of the major reasons for the lack of research is the lack of enabling technologies like, parts of speech taggers, gazetteers, and most importantly, corpora and annotated training and test sets. One of the first NER study of South Asian languages and specifically on Urdu was done by Becker and Riaz (2002) who studied the challenges of NER in Urdu text without any available resources at the time.
The by-product of that study was the creation of Becker-Riaz Urdu Corpus (2002).
Another notable example of NER in South Asian language is DARPA's TIDES surprise language challenge where a new language is announced by the agency to build language processing tools in a short period of time. In 2003 the language chosen was Hindi. Li and McCallum (2003) tried conditional random fields on Hindi data and reported f-measure ranging from 56 to 71 with different boosting methods. Mukund et al. (2009) used CRF for Urdu NER and showed f-measure of 68.9%.
By far the most comprehensive attempt made to study NER for South Asian and South East Asian languages was by the NER workshop of International Joint Conference of Natural Language Processing in 2008. The workshop attempted to do Named Entity Recognition in Hindi, Bengali, Telugu, Oriya, and Urdu. Among all these languages Urdu is the only one that has Arabic script. Test and training data was provided for each language by different organizations therefore the quantity of the annotated data varied among different languages. Hindi and Bengali led the way with the most amounts of data; Urdu and Oriya were at the bottom with the least amount of data. Urdu had about 36,000 thousand tokens available. A shared task was defined to find named entities in the languages chosen by the researcher. There are 15 papers in the final proceedings of NER workshop at IJCNLP 2008, all cited in the references section, a significant number of those papers tried to address all languages in general, but resorted to Hindi, where the most number of resources were available. Some papers only addressed specific languages like Hindi, Bengali, Telugu and one paper addressed Tamil. There was not a single paper that focused on only Urdu named entity recognition. The papers that tried to address all languages, the computational model showed the lowest performance on Urdu. Among the experiments performed at Named Entity Workshop on various Indic languages and Urdu, almost all experiments used CFR with limited success.

Saha et.al(2008) [5] describes the development of Hindi NER using ME approach. The training data consists of about 234k words, collected from the newspaper "Dainik Jagaran" and is manually tagged with 17 classes including one class for not name and consists of 16,482 NEs. The paper also reports the development of a module for semi-automatic learning of context pattern. The system was evaluated using a blind test corpus of 25K words having 4 classes and achieved an F-measure of 81.52%.

Goyal (2008) [6] focuses on building a NER for Hindi using CRF. This method was evaluated on test set 1 and test set 2 and attains a maximum F1-measure around 49.2% and nested F1-measure around 50.1% for test set 1 maximum F1-measure around 44.97% and nested F1-measure around 43.70% for test set 2 and F-measure of 58.85% on development set.

Saha et.al(2008) [7] has identified suitable features for Hindi NER task that are used to develop an ME based Hindi NER system. Two-phase transliteration methodology was used to make the English lists useful in the Hindi NER task. The system showed a considerable performance after using the transliteration based gazetteer lists. This transliteration approach is also applied to Bengali besides Hindi NER task and is seen to be effective. The highest F-measure achieved by ME based system is 75.89% which is then increased 81.2% by using the transliteration based gazetteer list.

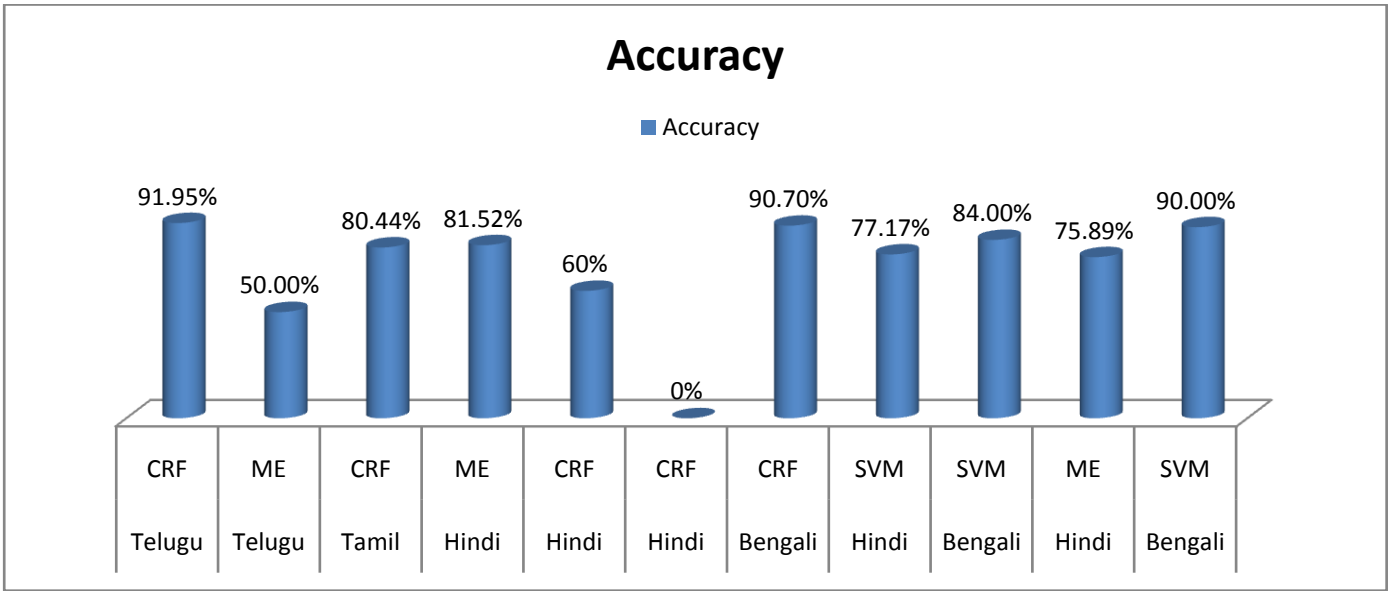
Li and McCallum (2004) [1] describes the application of CRF with feature induction to a Hindi NER. They discovered relevant features by providing a large array of lexical test and using feature induction to construct the features that increases the conditional likelihood. Combination of Gaussian prior and early-stopping based on the results of 10-fold cross validation is used to reduce over fitting.

Gupta and Arora (2009) [3] describes the observation made from the experiment conducted on CRF model for developing Hindi NER. It shows some features which makes the development of NER system complex. It also describes the different approaches for NER. The data used for the training of the model was taken from Tourism domain and it is manually tagged in IOB format.

David Nadeau *et al.* [12]. proposed a named-entity recognition (NER) system that addresses two major limitations frequently discussed in the field. First, the system requires no human intervention such as manually labeling training data or creating Gazetteers. Second, the system can handle more than the three classical named-entity types (person, location, and organization). They propose a named-entity recognition system that combines named entity extraction with a simple form of named-entity disambiguation. They use some simple yet highly effective heuristics, to perform named-entity disambiguation.

Deepti Chopra *et al*. [14]. have discussed about NER, Challenges in NER in the Indian languages, Performance Metrics and finally the methodology and the results. They have obtained F-Measure and accuracy of about 88.4% by performing NER in Punjabi using Hidden Markov Model (HMM).

VI. EXISTING WORK ON DIFFERENT INDIAN LANGUAGES



A. Fig 1: Different Approaches and Their Accuracy

REFERENCE

[1] A. Goyal , "Named Entity Recognition for South Asian Languages Jan 2008," in Proceedings of the IJCNLP-08 Workshop on NER for South and South-East Asian Languages, Hyderabad, India.

[2] Arindam Dey, Abhijit Paul, Bipul Syam Purkayastha," Named Entity Recognition for Nepali language: A Semi Hybrid Approach"(IJEIT) Working paper February 2014.
[3] Arindam Dey, Bipul Syam Purkayastha, "Named Entity Recognition using Gazetteer Method and N-gram Technique for an Inflectional Language: A Hybrid Approach" (IJCA) International Journal of Computer Applications, Vol. 84, 2013
[4] Anastasia Rita Widiarti, and Phalita Nari Wastu 2009, "Javanese Character Recognition Using Hidden Markov Model"World Academy of Science, Engineering and Technology 33.
[5] Anup Patel Ganesh Ramakrishnan Pushpak Bhattacharya, " Relational Learning Assisted Construction of Rule Base for Indian Language NER" ICON 2009 conference.
[6] Asif Ekbal, Rajewanul Hague, Amitava Das, Venkateswarlu Poka and Sivaji Bandyopadhyay 2008 "Language Independent Named Entity Recognition in Indian Languages" Proceedings of the IJNLP-08 Workshop on NER for South and South East Asian Languages, Hyderabad, India.
[7] Bal Krishna Bal, Prajol Shrestha, "A Morphological Analyzer and a Stemmer for Nepali", Working Paper 2004-2007.
[8] Bowen Sun "Named entity recognition Evaluation of Existing Systems" Norwegian University of Science and Technology Department of Computer and Information Science, Thesis.
[9] David Nadeau, Satoshi Sekine , "A survey of named entity recognition and classification" National Research Council Canada / New York University.
[10] David Nadeau, Peter D. Turney and Stan Matwin March 11 , 2011, "Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity" National Research Council Canada.
[11] Deepti Chopra, Sudha Morwal Dec 12, 2012, "Named Entity Recognition in Punjabi Using Hidden Markov Model", "International Journal of Computer Science & Engineering Technology (IJCSET)".
[12] Ijaz, M., Hussain, S., Corpus Based Lexicon Development, in the Proccedings of Conference on Language Technology. 2007
[13] Kashif Riaz," Rule-based Named Entity Recognition in Urdu" Proceedings of the 2010 Named Entities Workshop, ACL 2010.
[14] M. N. Karthik, Moshe Davis "Search Using N-gram Technique Based Statistical Analysis for Knowledge Extraction in Case Based Reasoning Systems" .
[15] M. Hasanuzzaman, A. Ekbal, and S. Bandyopadhyay, May 2009, "Maximum Entropy Approach for Named Entity Recognition in Bengali and Hindi, "International Journal of Recent Trends in Engineering, vol. 1.
[16] Padmaja Sharma, Utpal Sharma, Jugal Kalita May 2011, "Named Entity Recognition: A Survey for the Indian Languages".
[17] P. Srikanth, K. Murthy, Named Entity Recognition for Telugu, Workshop on NER for South and South East Asian Languages, IJCNLP 2008
[18] S. K. Saha, S. Sarkar, and P. Mitra January 2008, "A Hybrid Feature Set based Maximum Entropy Hindi Named Entity Recognition," in Proceedings of the 3rd International Joint Conference on NLP, Hyderabad , India.
[19] Suleiman H. Mustafa and Qasem A. Al-Radaideh 2004 "Using N-Grams for Arabic Text Searching" journal of the american society for information science and technology.
[20] W. Li and A. McCallum, Sept 2003 "Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction(Short Paper)," ACM Transactions on Computational Logic.
[21] Zubek, R. 2006. Introduction to Hidden Markov Models. In Rabin, S. (ed.), AI Game Programming Wisdom 3. Charles River Media, Hingham, MA.

