Various Statistical Techniques Used in NLP

Blossom Manchanda
Assistant Professor
Department of CSE
GNDEC Ludhiana

Vijay AnantAthavale
Director
Gulzar Group of Institutions
Khanna

ABSTRACT

In this research paper, author has explained various statistical techniques used in natural language processing. Statistical techniques explained are HMM based technique, conditional random field based technique, support vector machine based technique and N-gram based technique. These techniques are used in developing fundamental applications like morphological analyzer, part of speech tagger, phrase chunker and clause boundary identification etc. and advanced applications like grammar checker, spell checker, summarization system etc.

Keywords

NLP, Statistical techniques, rule based approach, HMM, SVM, CRF, N-gram

1. INTRODUCTION

Natural language processing is the branch of computer science that deals with processing of languages through computer. It is an inter discipline branch that deals with computer and languages. Various types of processing done on languages using computer includes speech processing, spell checker, grammar checker, summarization, dialogue processing etc.

2. INTRODUCTION TO STATISTICAL METHODS

In statistical techniques, frequency and probability process the natural languages. The probability is calculated from a training corpus. Many approaches can be used to implement statistical methods. These included Hidden Markov Model (HMM), Maximum Likelihood Estimation, Decision Trees, N-gram, Maximum Entropy, Support Vector Machines and Conditional Random Fields.

3. HIDDEN MARKOV MODEL (HMM)

HMM is one of the distinguished probabilistic models. It is used to work out on a number of different problems and also used in language processing problems. It is effectively utilized to find out most probable state sequence for a particular sentence. It works by assigning the joint probability to paired observation and label sequence. This model assigns the joint probability to paired observation and label sequence. Then the parameters are trained to maximize the joint likelihood of training sets. It is advantageous as its basic theory is elegant and easy to understand. Hence it is easier to implement and analyze. It uses only positive data, so they can be easily scaled. It has few disadvantages. In order to define joint probability over observation and label sequence HMM needs to enumerate all possible observation sequence. Hence it makes various assumptions about data like Markovian assumption i.e. current label depends only on the previous label. Also it is not practical to represent multiple overlapping features and long term dependencies. Number of parameter to be evaluated is huge. So it needs a large data set for training.

3.1 Basic Definitions and Notation

According to (Rabiner, 1989), there are five elements needed to define an HMM:

1. N, the number of distinct states in the model. For part-of-speech tagging, N is the number of tags that can be used by the system. Each possible tag for the system corresponds to one state of the HMM.

2. M, the number of distinct output symbols in the alphabet of the HMM. For part-of-speech tagging, M is the number of words in the lexicon of the system.

3. A = $\{a_{ij}\}$, the state transition probability distribution. The probability *a* is the probability that the process will move from state i to state j in one transition. For example in case of part-of-speech tagging, the states represent the tags, so $a_{ij}$is the probability that the model will move from tag t_i to t_j -- in other words, the probability that tag $t_j$follows t_i. This probability can be estimated using data from a training corpus.

4. B = {$b_j(k)$), the observation symbol probability distribution. The probability $b_j(k)$ is the probability that the k-th output symbol will be emitted when the model is in state j. For example in case of part-of-speech tagging, this is the probability that the word $W_k$will be emitted when the system is at tag t_j(i.e.,$P(W_k|t_j)$). This probability can be estimated using data from a training corpus.
5. ∏ = {$∏_i$}, the initial state distribution. $∏_i$is the probability that the model will start in state i. For example in case of part-of-speech tagging, this is the probability that the sentence will begin with tag ti. When using an HMM to perform part-of speech tagging, the goal is to determine the most likely sequence of tags (states) that generates the words in the sentence (sequence of output symbols). In other words, given a sentence V, calculate the sequence U of tags that maximizes *P (V/U).* The Viterbi algorithm is a common method for calculating the most likely tag sequence when using an HMM.

The presented model is a type of first order HMM, also referred to as bigram POS tagging. For example in case of POS-tagging problem presented Hidden Markov Model is composed of two probabilities: lexical (emission) probability and contextual (transition) probability (Samuelsson, 1996).

$$(t_1,\ldots,t_n)^* = \underset{t_1 \ldots t_n}{\operatorname{argmax}}\ P(t_1,\ldots,t_n)|(w_0,\ldots,w_n)$$

Using Baye’s law above equation can be rewritten as:

$$P(t_1,\ldots,t_n|w_1,\ldots,w_n) = P(t_1,\ldots,t_n) \text{ x } \frac{P(w_1,\ldots,w_n|t_1,\ldots,t_n)}{P(w_1,\ldots,w_n)}$$

$$(t_1,\ldots,t_n)^* = \underset{t_1,\ldots,t_n}{\operatorname{argmax}}\ P(t_1,\ldots,t_n) \text{ x } P(w_1,\ldots,w_n|t_1,\ldots,t_n)$$

$$(t_1,\ldots t_n)^* = \underset{t_1,\ldots t_n}{\operatorname{argmax}}\ P(t_1,\ldots t_n) \text{ x } P(w_1,\ldots w_n|t_1,\ldots t_n)$$

$$= \underset{t_1,\ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} (\underbrace{P(t_i|t_{i-1})}_{\text{Transition probability}} * \underbrace{P(w_i|t_i)}_{\text{Emission probability}})$$

4. MAXIMUM ENTROPY MARKOV MODEL (ME)

MaxEnt stands for Maximum Entropy Markov Model (MEMM). It is a conditional probabilistic sequence model. It can represent multiple features of a word and can also handle long term dependency. It is based on the principle of maximum entropy which states that the least biased model which considers all known facts is the one which maximizes entropy.Each source state has an exponential model that takes the observation feature as input and output a distribution over possible next state. Output labels are associated with states. The large dependency problem of HMM is resolved by this model. Also, it has higher recall and precision as compared to HMM. The disadvantage of this approach is the label bias problem. The probabilities of transition from a particular state must sum to one. MEMM favors those states through which less number of transitions occurs. The ME technique construct a model based on constraints. In order to build this model feature functions are defined. The probability distribution that satisfy the constraints and that makes no other assumptions has maximum entropy, is unique (Berger et. al. 1996). It can be expressed as:

$$\Pr(l|c) = \frac{1}{z(c)} \exp\left(\sum_{j=1}^{k} \lambda_j f_j(l,c)\right)$$

Here z(c) is a normalized constant. The problem of estimating parameter$\lambda_j$can be solved by using Generalized Kerative Scaling (Darroch and Ratcliff, 1972) algorithm. ME based approach has been used for Hindi (Ankit dalal et al., 2009), Bengali (Asif Iqbal et al., 2008)

5. CONDITIONAL RANDOM FIELD MODEL (CRF)

Conditional Random Fields (CRF) is a relatively new mathematical model thatmay be employed to solve sequence labelling problems (Lafferty et al. 2001).CRF is a probabilistic model which is a statistical based approach that predicts sequences of labels or tags for the given input data. CRFs are undirected graphical models, which are also known as random field. These are used to calculate the conditional probability p(x|y) of a possible output nodes y=(y1,…,yn) corresponding to the input or observation x=(x1,…,x2). The general expression for CRF is

PΛ(s|o)= $\frac{1}{z_0}$exp($\sum_{t=1}^{T}\sum_k \lambda_k f_k(s_t$ 1,s_t, o, t))

Where $f_k(s_t$ 1,s_t, o, t) is a feature function and its weight λ_k is learned through training.

In natural language processing main features used are neighboring words and word bigrams, prefixes and suffixes, capitalization, membership in domain-specific lexicons and semantic information from source. The CRF is applied to a variety of natural language processing domains such as text processing, computer vision, and bioinformatics. Different feature functions are used in different applications like for part of speech tagging most commonly used feature are context base feature (previous n word, next n word), POS tag related features (previous n tags, next n tags), orthographic features based upon language (prefix, suffix of the word) and many more. More the number of features more will be the accuracy of application. .CRF based approach has been used for Manipuri (ThoudamDoren Singh et al., 2008), Bengali (Asif Iqbal et al., 2008) and Gujarati (Chirajpatel et al., 2008). Also this model has been successfully applied to various Natural Language Processing (NLP) tasks, including Part-of-Speech tagging of English. The list of successful applications also includes processing of Polish, e.g., concept-tagging of spoken language corpora (Lehnen et al. 2009), named entity recognition (Marcinczuk and Janicki 2012) and NP chunking (Radziszewski and Pawlaczek 2012).

6. SUPPORT VECTOR MACHINE (SVM)

These were first introduced by Vapnic (refer blue file #1 25, 26). It is suitable for solving two class pattern recognition problems in natural language processing. In natural language processing, it is mainly used for text categorization. For example in case of POS tagging different features are extracted from the corpus. These features include Context word feature, word suffix, word prefix, POS information and lexicon features. SVM based approach has been used for developing the part of speech tagger for Malayalam (Antony P et al., 2010), Bengali (Ekbal A et al., 2008), Manipuri (ThoudamDoren Singh et al., 2008), Tamil (Dhanalakshmi V et al., 2009), Kannad (Antony P. and Soman K, 2010)[3].

7. N-GRAM

It is a probability based statistical technique. The basic requirement to implement this technique is an annotated corpus. Most of the Indian languages lack this basic resource. A word is assigned a tag based upon the frequency or probability of that tag to occur with that word. The frequency or probability of that tag to occur with a word is calculated from a pre annotated corpus. This probability is used to assign the correct tag to the word in the tested corpus. For example if N is taken as two then it becomes B-gram. Then probability values are pre calculated from a training corpus. The accuracy of this technique increases with increase in training corpus.

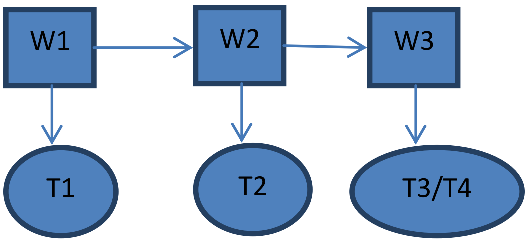

Fig:-1 bigram model

As shown infigure 1, first two words have single word class and the third word has an ambiguity i.e. has two word classes. In bigram approach all the combinations of ambiguous word classes with previous word class are created i.e. T2_T3 and T2_T4. Then each word class pair is assigned a probability from bigram probability file. From these two word class pairs, the one having maximum probability is selected and the word classes from this pair are assigned to respective words. N-gram HMM based approach has been widely used for many languages like Bangla (Hammad Ali, 2010), Malayalam (Manju k et al., 2009), Hindi, Bengali, Marathi and Telugu (Kamal Sarkar et al., 2013) and Hindi (Nisheeth Joshi et al., 2013).

8. GENETIC ALGORITHM (GA)

'Genetic Algorithms' are capable of being applied to an enormously wide range of problems. The concept of GA was given by John Holland. Some of the major applications of these algorithms are Image Processing, Query Optimization, Natural Language Processing, Task Scheduling, Biobank queries, Clustering, Game Theory, Artificial Intelligence, Aeronautics, etc. In general, these types of algorithms aim for searching better solution from a number of available solutions[1][2][14][15]. Evolutionary computing involving interaction of agents can model complex behavior in language, such as the development of language strategies, or the induction of grammars. Grammars that are induced from agent interactions learn and adapt to new samples, and can in principle discover grammatical structures (utterances) that have not yet been uttered. On a more concrete level, there is a need in most NLP application areas to be able to dynamically adapt to new language domains (sublanguages), and evolutionary algorithms is one way of doing this. It would be interesting to see an evolutionary algorithm applied to rebuilding and reconfiguring a translation model in statistical machine translation dynamically as new training instances and corrections are being provided by the user. In facilitating the emergence of structures from data, and the modeling of language development and etymology, evolutionary computation has shown a lot of promise, helped by the present-day increase in available data and computing power. As computer resources grow, more agents and more generations can be introduced. As computational linguistics plays a bigger part in historical linguistics, evolutionary algorithms should have a part in this.

9. CONCLUSION

Statistical techniques play an important role in processing of languages. Almost all the natural language processing applications can be developed by using one or combining more than one statistical technique. A list of statistical techniques used to process natural language processing and various applications that can be developed using these techniques is tabulated in table 1.

Table 1: Statistical techniques used in various NLP resources.

Sr. No.	Statistical Approach	NLP Resources
1	HMM	POS Tagging, clause identification, phrase chunking
2	SVM	POS Tagging, name entity recognition (NER)
3	CRF	POS Tagging, clause identification, phrase chunking, anaphora resolution, spell checking
4	N-gram	POS Tagging, name entity recognition (NER), grammar checking
5	Maximum Entropy	POS Tagging
6	Genetic Algorithm	Syntactic analysis, grammar checker, spell checker

REFERENCES

[1]. Manik Sharma, Gurvinder Singh, Rajinder Singh and Gurdev Singh. 2015. “Analysis of DSS Queries using Entropy based Restricted Genetic Algorithm”. Applied Mathematics & Information Science. Volume 9 Number 5.

[2]. Manik Sharma, Gurvinder Singh, Rajinder Singh. “Design and Analysis of Stochastic DSS Query Optimizer in a Distributed Database System”. Egyptian Informatics Journal. doi:10.1016/j.eij.2015.10.003

[3]. Gimenez J. and Marquez L., “Fast and accurate part-of-speech tagging: The SVM approach revisited,” Proc. Recent Advances in Natural Language Processing III, pp.153-162, 2004.

[4]. Kashyap D. and Josan G., ”A Trigram Language Model to Predict Part of Speech Tags Using Neural Network,” Springer LNCS 8206, China,pp. 513-520, 2013

[5]. Ekbal A. and Bandyopadhyay S., “Part of speech tagging in Bengali using support vector machine,” Proc. IEEE International Conference on Information Technology, Bhubneswar, India,pp. 106-111, 2008.

[6]. Antony P. and Soman K., “Kernel based part of speech tagger for kannada,” Proc. IEEE International Conference on Machine Learning and Cybernetics (ICMLC), Qingdao, vol. 4, pp.2139-2144, 2010.

[7]. Antony P., Mohan S., Soman K., “SVM based part of speech tagger for Malayalam,” Proc. IEEE International Conference on Recent Trends in Information, Telecommunication and Computing (ITC), Kerala , India, pp. 339-341,2010.

[8]. Charniak E., Hendrickson C., Jacobson N., and Perkowitz M., “Equations for part-of-speech tagging,” Proc. 11th National Conference on Artificial Intelligence, Washington, D.C., pp.784-784, 1993

[9]. Ratnaparkhi A., “A maximum entropy model for part-of-speech tagging,” Proc. Empirical methods in natural language processing, Philadelphia, PA, pp. 133-142, 1996.

[10].Jurafsky, D., & Martin, J. H. (2000). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, NJ: Prentice-Hall.

[11].Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39–71.

[12].J.N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics, 43(5):1470–1480.

[13]. Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In Eric Brill and Kenneth Church, editors, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey.

[14]. Manik Sharma, Gurvinder Singh, Rajinder Singh. 2016. "Statistical Analysis of DSS Query Optimizer for a Five Join DSS Query". International Journal of Computer Applications. 141(6):1-4. 10.5120/ijca2016909627

[15]. Manik Sharma, Gurvinder Singh, Rajinder Singh. 2013. "Design and comparative analysis of DSS queries in distributed environment". IEEE International Conference on Computer Science and Engineering (ICSEC 2013). Doi: 10.1109/ICSEC.2013.6694756

[16]. AdwaitRatnaparkhi. 1997. A simple introduction tomaximum entropy models for natural language processing.Technical Report 97-08, Institute for Researchin Cognitive Science, University of Pennsylvania,May.