



PARTS OF SPEECH TAGGING USING VITERBIALGORITHM

N.Jahnavi¹, P.V.Parimala², N.Venakata Vardhan³, K.Uday Kishore⁴, L.Ravinandan⁵ and G.Rajendra Kumar ⁶

^{1,2,3,4,5}-Final year Bachelor of Technology(B.Tech)Students,⁶-Professor

Department of Information Technology, Vignan's Institute of Information Technology(VIIT)

ABSTRACT

Part-of-speech (POS) tagging is a popular Natural Language Processing process that refers to categorizing words in a text in correspondence with a particular part of speech, part of speech (POS) tagging is the ability to computationally verify that the POS of a word is activated by its use in an explicit context. POS tagging is changing progressively fashionable lately. Text to speech, syntactical analysis, and artificial intelligence all get pleasure from POS tagging, which is a good type of preprocessing tagging. once it involves POS taggers, they have to be well- versed in the term. However, if the quantity of words is inflated to 1,000,000, users are unable to finish the POS tag. we have a tendency to gift the Viterbi methodology during this study to help computers in tagging lexical classes effectively. The Viterbi formula uses dynamic programming to unravel issues. As we have a tendency to all understand, the word is kind of sensitive to its placement. The word's POS is connected to the words around it. we have a tendency to run simulations to examine however Viterbi Algorithms operate in POS taggers and calculate accuracy.

INTRODUCTION

Part-of-Speech (POS) (noun, verb, and preposition) will facilitate in understanding the means of a text by characteristics however completely different word area units employed in a sentence. POS will reveal tons of knowledge concerning near words and the syntactical structure of a sentence. POS tagging is the method of distributing a POS marker (noun, verb, etc.) to every word in Associate in Nursing input text. The input to a POS tagging algorithmic rule could be a sequence of tokenized words and a tag set (all potential POS tags) and also the output could be a sequence of tags, one per token. Words within the West Germanic area unit are ambiguous as a result of they need multiple POS. as an example, a book may be a verb (book a flight for ME) or a noun (please offer me this book). POS tagging aims to resolve those ambiguities. In such POS tagging task, we've got evident values delineated by the sentences and their words and that we have hidden states delineate by the tags like 'noun', 'verb', 'adjective', 'pronoun', etc... that we wish to connect to every word.

Back in elementary school, we learned the differences between the various parts of speech tags such as nouns, verbs, adjectives, and adverbs. Associating each word in a sentence with a proper POS (part of speech) is known as POS tagging or POS annotation. POS tags are also known as word classes, morphological classes, or lexical tags. Back in the day, the POS annotation was manually done by human annotators but being such a laborious task, today we have automatic tools that are capable of tagging each word with an appropriate POS tag within a context.

LITERATURE SURVEY

A stochastic approach includes frequency, probability, or statistics. The simplest stochastic approach finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the unannotated text. But sometimes this approach comes up with sequences of tags for sentences that are not acceptable according to the grammar rules of a language. One such approach is to calculate the probabilities of various tag sequences that are possible for a sentence and assign the POS tags from the sequence with the highest probability. Hidden Markov Models (HMMs) are probabilistic approaches to assigning a POS Tag. Hidden Markov models are able to come through >96% tag accuracy with larger tag sets on realistic text corpora. Hidden Markov models have additionally been used for speech recognition and speech generation, artificial intelligence, sequence recognition for bioinformatics, human gesture recognition for pc vision, and more...

Methodology

Identifying a part of speech tags is far a lot of difficult than merely mapping words to their part of speech tags. this is often a result of POS tagging isn't one thing that's generic. it's quite adorable for one word to possess a totally different special unique distinct part of speech tag in numerous sentences supported in different contexts. that's why it's not possible to possess a generic mapping for POS tags. Let us assume a finite set of words V and a finite sequence of tags K . Then the set S will be the set of all sequences, tags pairs $\langle x_1, x_2, x_3 \dots x_n, y_1, y_2, y_3, \dots, y_n \rangle$ such that $n > 0 \forall x \in V$ and $\forall y \in K$.

For any $\langle x_1 \dots x_n, y_1 \dots y_n \rangle \in S$,

$$p(x_1 \dots x_n, y_1 \dots y_n) \geq 0$$

Given a generative tagging model, the function that we talked about earlier from input to output becomes

$$f(x_1 \dots x_n) = \arg \max_{y_1 \dots y_n} p(x_1 \dots x_n, y_1 \dots y_n)$$

Thus, for any given input sequence of words, the output is the highest probability tag sequence from the model. Having defined the generative model, we need to figure out three different things:

1. How exactly do we define the generative model probability $p(\langle x_1, x_2, x_3 \dots x_n, y_1, y_2, y_3, \dots, y_n \rangle)$
2. How do we estimate the parameters of the model, and
3. How do we efficiently calculate?

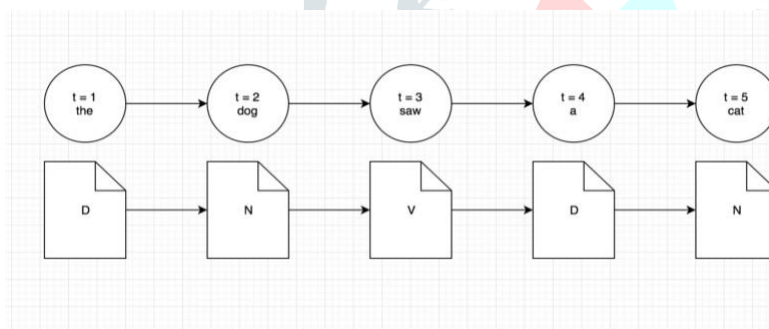
- **Transition probability** is defined as the probability of a state “s” appearing rightafter observing “u” and “v” in the sequence of observations. $e(x|s)$
- **Emission probability** is defined as the probability of making an observation x giventhat the state was s.

Then, the generative model probability would be estimated as

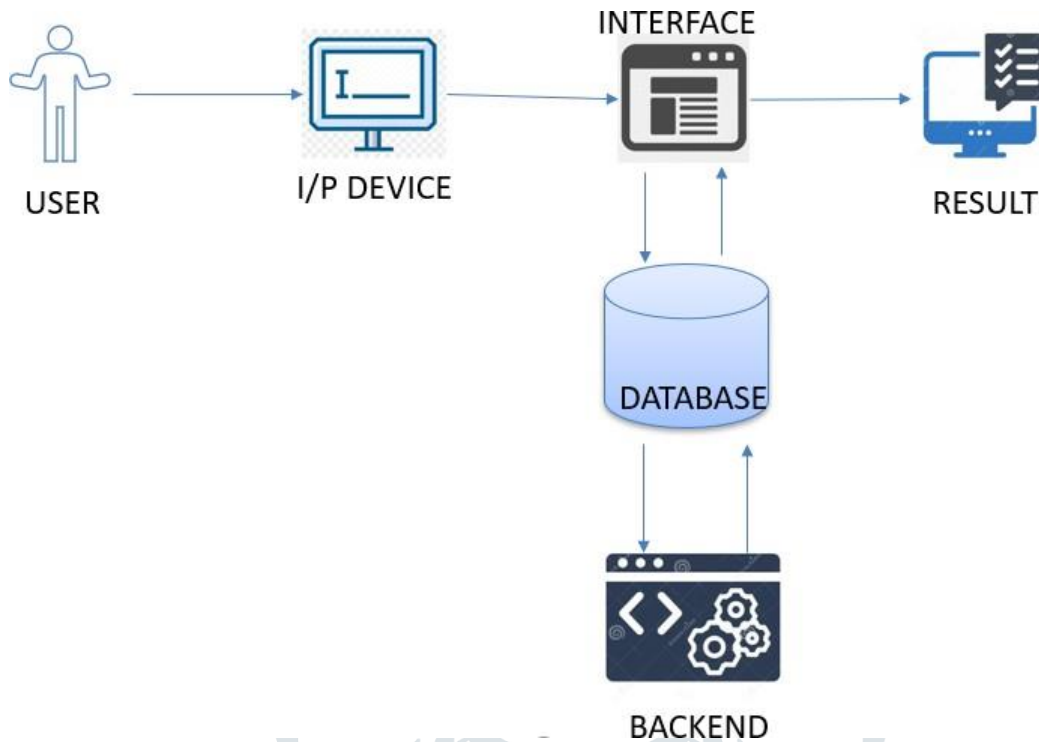
$$p(x_1 \dots x_n, y_1 \dots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-2}, y_{i-1}) \prod_{i=1}^n e(x_i | y_i)$$

The Naive Bayes formula is used in this formula. The letter x indicates the word, while the letter y signifies the portion of the speech. Then $p(y|x)$ denotes the likelihood that this word belongs to a specific category, which is known. However, it is difficult to locate. The Bayes formula is then used to calculate $p(y|x)$ from $p(x|y)$, which refers to the likelihood of a specific word knowing the part-of-speech of the word, and $p(y)$ to derive $p(y|x)$

In the part of the speech tagging problem, the states would be represented by the actual tags assigned to the words. The words would be our observations. The reason we say that the tags are our states is that in a Hidden Markov Model, the states are always hidden and all we have are the set of observations that are visible to us. Along similar lines, the sequence of states and observations for the part of the speech tagging problem would be



ARCHITECTURE



VISUALISATION

To illustrate the internal processing we have considered the sentence "when are you coming".

Table 1

PROCESSING SENTENCE : When are you coming

The below image shows the probability matrix for words of the sentence and part of speech after applying the algorithm. Log is applied to probabilities to get an accurate value. For “when” WRB has the highest probability. And then for “are” VBP has the highest probability among other parts of speech tags. for “you” PRP which is a personal pronoun has the highest probability. And then for “coming” VBG has the highest probability.

	when	are	you	coming
--init--	-27.692103	-43.065634	-50.323901	-55.437403
CC	-33.984561	-37.855207	-44.014869	-48.019710
CD	-34.830583	-40.000485	-46.891039	-52.449212
DT	-36.441145	-44.905922	-47.188448	-53.589786
EX	-27.386550	-42.760081	-37.768254	-48.223095
FW	-24.900048	-38.866970	-40.623091	-52.645348
IN	-36.812681	-40.203277	-49.278589	-49.652227
JJ	-35.860597	-40.565149	-47.633377	-42.224020
JJR	-29.994563	-37.766692	-44.108968	-48.886055
JJS	-28.986167	-36.065399	-41.377970	-47.521027
LS	-21.787184	-37.160715	-44.418982	-49.532484
MD	-32.200921	-47.574452	-46.826018	-49.068155
NN	-37.411076	-40.111658	-46.885142	-51.889045
NNP	-36.663441	-34.328622	-46.457893	-53.210513
NNPS	-29.613911	-44.987443	-40.347515	-49.064912
NNS	-35.815647	-39.437228	-43.007411	-51.040060
PDT	-25.755943	-41.129475	-48.387742	-53.501243
POS	-31.962991	-40.427768	-46.077396	-52.799537
PRP	-33.350640	-38.771847	-27.745300	-50.855945
PRP\$	-31.894422	-47.267954	-53.304019	-59.639722
RB	-34.498493	-38.282129	-43.537923	-49.659281
RBR	-28.795542	-37.260319	-44.518586	-47.687034
RBS	-26.132423	-41.505955	-41.855467	-46.968969
RP	-29.605731	-35.506481	-44.636128	-49.749629
SYM	-22.505526	-37.879057	-45.137324	-50.250826
TO	-33.847286	-40.521136	-49.570329	-49.975291
UH	-23.342211	-38.677720	-45.974009	-44.178756
VB	-34.182305	-37.699314	-47.603663	-48.128223
VBD	-34.427488	-39.022042	-48.764987	-47.769490
VBG	-33.029471	-38.199373	-46.556178	-36.366396
VCN	-33.627101	-38.379281	-46.132228	-49.222894
VBP	-32.684589	-22.631798	-45.843606	-47.401834
VBZ	-33.785112	-37.832035	-47.899517	-48.366905
WDT	-30.555754	-45.929285	-39.872915	-46.376012
WP	-29.369515	-44.743047	-39.148135	-44.850469
WP\$	-24.298168	-39.671699	-46.929966	-52.043468
WRB	-15.373532	-37.640735	-42.502015	-43.845984

EXCEPTION HANDLING

If the given sentence is not present in vocabulary, then the parts of speech of the sentence are not predicted and the text is not present in vocabulary is printed on the results page.

CONCLUSION

As we mentioned POS tagging, is a text process technique to extract the connection between near words during a sentence. POS tagging resolves ambiguities for machines to know the language. In informal systems, an oversized range of errors arises from the language understanding module. POS tagging is one technique to reduce those errors in informal systems. The Viterbi method uses dynamic programming to compute the likelihood of a word in every conceivable POS and choose the best one as the final POS tagger. We can simply determine that the noun is clearly identified based on simulation data.

REFERENCES

- [1] Gallier, "Evaluating Natural Language System," Springer, vol. A24, pp. 529–551, April 1996.
- [2] GD Forney, "The Viterbi algorithm," Proceedings of the IEEE, 1973.
- [3] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," Proc. of the IEEE, vol. 61, no. 3, pp. 268-278, Mar. 73.
- [4] G. Unger Boeck, "Channel Coding with Multilevel/Phase Signals", IEEE Trans. on Inf. Theory, vol. IT-28, no. 1, pp. 55-67, Jan. 82.
- [5] J. B. Anderson, T. Aulin, C.-E. Sundberg, Digital Phase Modulation, New York: Plenum Publishing Co., 1986.
- [6] Alebachew Chiche, Betselot Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches", Jan 2022.
- [7] Laurent Veyssier, "Part-Of-Speech-tagging-NLP-task" Oct 2020.
- [8] Sachin Malhotra and Divya Godayal, "An introduction to part-of-speech tagging and the Hidden Markov Model, June 2018.
- [9] Deep Mehta, "Part of Speech Tagging- POS Tagging in NLP", Jan 2021.
- [10] Susan Li, "Parts of Speech Tagging with Hidden Markov Chain Models", May 2018.
- [11] Divyapushpalakshmi M, Ramalakshmi R. An efficient sentimental analysis using hybrid deep learning and optimization technique for Twitter using parts of speech (POS) tagging. Int J Speech Technol. 2021;24(2):329–39.
- [12] Kumawat D, Jain V. POS tagging approaches: a comparison. Int J Comput Appl. 2015;118(6):32–8.
- [13] Jyoti Singh, Nisheeth Joshi and Iti Mathur, "Marathi Parts-of-Speech Tagger Using Supervised Learning" in Intelligent Computing Networking and Informatics, New Delhi: Springer, pp. 251-257, 2014.
- [14] Lv C, Liu H, Dong Y, Chen Y. Corpus based part-of-speech tagging. Int J Speech Technol. 2016;19(3):647–54.
- [15] Demilie WB. Analysis of implemented part of speech tagger approaches: the case of Ethiopian languages. Indian J Sci Technol. 2020;13(48):4661–71.