Computer Program for Counting the Part of Speeches, Text Narrations by using Secondary Data Algorithm Techniques

Pradeep Kumar
Research Scholar
Dept. of Information Technology
SSJ Campus Almora

Sumit Khulbe
Assistant Professor
Dept. of Information Technology
SSJ Campus Almora

H S Dhami
Director ICT & Head
Mathematics
Kumaun University,
SSJ Campus Almora

ABSTRACT
The given paper is an attempt in the direction of generation of Primary data by refinement of secondary data available at www.Gotaggersoft.in . The Gotagger software only distinguishes the narrations in text but does not count them. To overcome this limitation, we have embedded our work with Gotagger and have used portability technique for identifying the narrations. An application of dialogue systems for developing computer programs has been also described and discussed in this paper.

Keywords
Advanced designing and algorithm (ADA), recursive functions.

1. INTRODUCTION
A narration is a constructive format (as a work of speech, writing, song, film, television, video games, photography or theatre) that describes a sequence of non-fictional or fictional events. The word derives from the Latin verb narrare, "to recount", and is related to the adjective gnarus, "knowing" or "skilled". The word "story" may be used as a synonym of "narrative". It can also be used to refer to the sequence of events described in a narrative. A narrative can also be told by a character within a larger narrative. In corpus linguistics, part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags.

One of the basic tools and components necessary for any robust Natural Language Processing infrastructure of a given language, is Part-Of-Speech tagging (POST) also known POS-tagging or just Tagging as cited in {[3], [2]}. It is considered as one of the basic tools needed in speech recognition, natural language parsing, information retrieval and information extraction. Moreover, POST is also considered as first stage for analyzing and annotating corpora. POST is the process by which a specific tag is assigned to each word of a sentence to indicate the function of that word in the specific context we can cite the work of Jurafsky & Martin [15]. Different English tagging have recently emerged, some of them are developed by companies (Xerox, Sakhr, RDI) as commercial products, while others are a result of research efforts in the scientific community. We can find their references in the works of {[10], [17], [7],[4],[25]}. Among these works, Freeman and Khoja [10] have combined statistical and rule-based techniques and used a tagset of 131 basically derived from the BNC English tagset. This work is based on the Brill tagger and uses a machine learning approach. A tagset of 146 tags, based on that of Brown corpus for English is used. The work of Maamouri M et al. [17] is based on the automatic annotation output produced by the morphological analyzer of Buckwalter [24] it achieved an accuracy of 96%. Again the work of Diab M. et al.[7] uses Support Vector Machine (SVM) method and the LDC's POS tag set, which consists of 24 tags. Banko et al [4] Presents a

HMM tagger that exploits context on both sides of a word to be tagged. The need for generation of Secondary data review and analysis is a form of research and data compilation that is demanding and time-consuming; however, without proper citation (i.e., author, date, title) of materials that we used, our work will often be disregarded as it will only have limited use by those who wish to follow in our footsteps. Secondary data is generally referred to as outcome data. This is because secondary data generally describe the condition or status of phenomena or a group; however, these data alone do not tell us why the condition or status exists. This limitation can be overcome in two ways.

First, it can be overcome by using information from case studies and other research to fill in the gaps. Second, analysis of additional key data and indicators can help us acquire more explanation as to why a problem exists.

Running the genetic algorithm with a selected number of individuals to be analyzed. The higher the number of individuals to be tested, the higher the required computational time and the higher the confidence in the optimization results.

The present work describes a system for primary and secondary data summarization of texts by involving portability concepts in order to evaluate the degree of significance of words, groups of words and sentences. Gotagger is one particularly fruitful source of user-created content, and a flurry of recent research has aimed to understand and counting these data as given in the works of {[1],[5],[16],[21],[6],[23]}.

However, the bulk of this work eschews the standard pipeline of tools which might enable a richer linguistic analysis; such tools are typically trained on news text and have been shown to perform poorly on Twitter this work cited in Finin Tim et al. [9]. One of the most fundamental parts of the linguistic pipeline is part-of-speech (POS) tagging, a basic form of syntactic analysis which has countless applications in NLP.

Most POS taggers are trained from tree banks in the newswire domain, such as the Wall Street Journal corpus of the Penn Treebank Mitchell et al. [18]. Tagging performance degrades on out-of-domain data, and Tagger poses additional challenges due to the conversational nature of the text, the lack of conventional orthography, and 140- character limit of each message. Again we conclude our result with primary software (GoTagger) and reverse the string for generalizing the work by comparing it with 140- characters limit.

In this paper, we produce an English POS tagger that is designed especially for Tagger data. Our contributions are as follows:

- We developed a POS tagset for Tagger,
- We manually tagged 32 Narrations,
- We developed features for Tagger POS tagging and conducted experiments to evaluate them,
- We provide our corpus and maintained POS Tagger.

This was made possible by two things:

(1) An annotation scheme that fits the unique characteristics of our data and provides an appropriate level of linguistic detail, and (2) a feature set that captures Tagger-specific properties and exploits existing resources such as tag dictionaries and phonetic normalization.

Our tagger is a conditional random field given by in the work of John Lafferty et al. [14] enabling the incorporation of arbitrary local features in a log-linear model. Our base features include: a feature for each word type, a set of features that check whether the word contains digits or hyphens, suffix features up to length 3, and features looking at capitalization patterns in the word. We then added features that leverage domain specific properties of our data, unlabeled in-domain data, and external linguistic resources. Since secondary Tagger includes many alternate spellings of words, we used the Metaphone ADA algorithm cited by Philips et al. [19] to create a coarse phonetic normalization of words to simpler keys. Metaphone consists of 19 rules that rewrite consonants and delete vowels.

The software can be used as integration for other systems or for purposes different from relation extraction and counting Multi-Way Classification of Semantic Relations between Pairs of Taggers Hendricks et al. {[11],[12]}. A different version of this system based on Part-Of-Speech counting has been previously used for the automatic annotation of three general and separable semantic relation classes (taxonomy, location, association).

The present work is an answer to the basic inquisition that whether computer software can have any answer to this language phenomenon especially in the context of the need of the students, who while using English language face difficulty in applying grammatical techniques in syntactic theory, like that of counting and tagging of narrations from large text. We are making efforts for formulation and then development of program for counting and tagging of narrations from a huge corpus.

2. OUTLINE OF PROCEDURE TO BE ADOPTED

We set out to develop a POS inventory for Tagger that would be intuitive and informative while at the same time simple to learn and apply so as to maximize tagging consistency within and across narrations. Thus, we sought to design a corpus tag set that would capture standard parts of speech (noun, verb, etc.) as well as categories for token varieties seen mainly in secondary data. The 36 keywords have been extracted and then they should be tagged by secondary data to formed primary data as to count the different narrations in text.

In knowledge discovery in a text database, counting a subset of information is highly relevant to a user's query is a critical task. In a broader sense, this is essentially identification of certain personalized patterns that drives such applications as, customized text summarization and automated question answering. A useful first step in document summarization is the selection of a small number of meaningful sentences from a larger text and it is a technique used for automatic summarization. This work can be accomplished with the help of a technique in which every word within a document, the scoring is performed based on a plurality of parameters which are adjusted through training prior to use of the method for counting and tagging. [22] seeks to use spoken language processing techniques to count the keywords from news, using an encyclopedia and newspaper articles as a guide for relevance, [20] set out to evaluate two lexical resources: the EDR electronic dictionary, and Princeton University's freely available Word Net. [13] Proposes that linguistic properties of texts will yield higher quality keywords and better retrieval, and examines a few different methods of incorporating linguistics into keyword tagging. Our major work was whether strong differences between groups in narratives with impact of secondary data.

3. MODUS OPERANDI FOR COMPUTER PROGRAM

With a general concept of the topic now in mind, you would next consult as many different secondary sources as possible to see what has already been written on the topic, at different times and from different points of view, by other scholars ('experts' on the topic). "Secondary" sources are thus works written on the topic in question by other researchers, whose work has been based on Primary sources after consultation with the Secondary sources on the topic which had existed at the time. The "Review of the Literature" component of full research papers is precisely this wide-ranging review of what all known secondary sources currently say about a given topic, as the foundation for the "new" information you plan to provide in your research.

The overall selected words or narratives of an individual j (a particular narrations) is assigned according to its rank r(j), which is defined from the number of assigned narratives by which function F(j) is a domain, and always increased 1 and call by recursive function as defined by secondary data. Thus, the individuals have rank equal to one and the maximum value of the rank is equal to the text size.

We used the Rank method (for a particular text) language pseudo -code is the following:

For i=1 to N

Rank (i) =1

For m=1 to N

If ((F 1 (i) - F 1 (m) <=0) and (F 2 (i) - F 2 (m) <=0) and

(F 3 (i) - F 3 (m) <=0) and (F 4 (i) - F 4 (m) <=0))

Individual is dominated

Rank (i) = Rank (i) + 1

next m

next i

The preceding narration should come first whose Rank is checked by the user.

Where F1, F2, F3, F4 are the different narratives in a text defined by secondary data.

After computing the rank of each narrative, a portion of the text in inverse proportion to its rank is assigned to the individual word. In this way, selection is inclined to prefer no domain and function itself recursive.

We always reanalyzed the concept of ADA by tagging the word/narratives in a sequence of above format and then recurs the function F1…… FN and count the narratives in a text either by ascending or descending order.

4. SYSTEM DESCRIPTION

Our Primary data is a Part-Of-Speech (POS) Context Counter. Given as input a plain text with Part-Of-Speech and end-of-sentence markers annotated it outputs a numerical feature vector that gives a representation of a sentence. For Part-Of-Speech and end-of sentence annotation we used ADA, a tool of computer science that showed state-of-the-art performance for POS tagging given by Emanuele Pianta et al. [8].

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.

Features in this work can be counted for specific tasks; in this case 12 features we used in total. They are:

1. Number of prepositions in sentences.

2. Number of nouns and proper names in sentences.

3. Number of conjunctions in sentence.

4. Number of adjectives in sentence.

5. Number of pronouns in sentence.

6. Number of punctuations in sentence.

7. Number of negative particles in sentence.

8. Number of words in the context between the first and the second entity.

9. Number of verbs in the context between the first and the second entity.

10. Patterns (from, in, on, by, of, to).

11. POS of entity 1 (noun, adjective, other).

12. POS of entity 2 (noun, adjective, other).

Example for narrations in primary data:-

WORD	TAG
heat	verb (noun)
water	noun (verb)
in	prep (noun, adv)
large	adj (noun)
vessel	noun

5. FOUNDATION OF THE COMPUTER PROGRAM

This work is an attempt to develop software based on computer tools for the counting of narrations from the text form without altering its sense. We have utilized an organized process in which consecutive versions are obtained by employing changes to the syntax definition of a language. We have demonstrated that formulation of the syntax definitions of programming languages and the processes by which these are extended and maintained improve the quality of syntax definitions and it results in the automatic generation of related tooling. In order to accomplish the work, we have made the use of software engineering concepts, prominent among them being:

Computer languages and specification formalisms

Tool support software-Advanced Integrated Development Environments (IDEs) like Microsoft Visual Studio, MS-Office Packages and VB.Net.

Process of software development in order to be reusable and for the generation of many artifacts from existing data.

The hardware requirements of this project are- 40GB Hard Disk Drive (HDD), 256MB Random Access Memory (RAM), all the buses which are generally used and simple Intel mother board via chip. C++ has been used as File Handling to help over the several database files. The developed software requires only 50 MB space in the unzipped category and can

run efficiently even in 17'' Color Monitor. We have used Window XP Operating System to develop this project.

6. DESCRIPTION OF THE COMPUTER PROGRAM

```
Public Function getWordCount (ByVal InputString As String) As Integer

ReturnCount(System.Text.RegularExpressions.Regex.Replace(InputString,"\s+",  Space(1))).Length

  End Function

        'Elimination of key words and their replacement:

Public Function StripStopWords (ByVal As String) As String

Dim StopWords As String = ReadFile("C:\Users\jaimiechin\Desktop\stopwords.txt").Trim

Dim StopWordsRegex As String = Regex.Replace(StopWords, "\s+", "|") ' about|after|all|also etc.

StopWordsRegex = String.Format("\s?\b(?:{0})\b\s?", StopWordsRegex)

Dim Result As String = Regex.Replace(s, StopWordsRegex, " ", RegexOptions.IgnoreCase) ' replace each stop word with a space

Result = Regex.Replace(Result, "\s+", " ") ' eliminate any multiple spaces

'Return Result

End Function

Sub main()
```

Making the use of information retrieval, we have modulated all the information related to counting the narrations and tagged the English language sentences. That particular database has been maintained depending upon user's requirement. The demodulation process has helped us in retrieving the related information from the associated database.

With the help of this software we can also specified the taggers. If we are asked to count the sentence and tagged them then we have to simply write the sentence into the terminal software and click into desired buttons for counting and tagging.

7. EXPERIMENTAL RESULTS OF THE PROPOSED SCHEME

The concept of generating program source code by means of a dialogue initiated by us for the first time in this context, involves combining strategies with system and user initiative. The strategy with system initiative safely navigates the user, whereas the strategy with user initiative enables a quick and effective creation of the desired constructions of the source code and collaboration with the system using obtained knowledge to increase the effectiveness of the dialogue. Novice programmers can also use the described system, which was initially developed for visually impaired users, as a tool for learning programming languages.

This section deals with the background details of the working of the software and for this purpose we have formed a link of the full syntax and coding in the mail entitled research with subject-Project4, bearing the name

Program2010@rediffmail.com

with password software. The present invention sets forth a method and an arrangement for different-different word counting processing and can automate the process of adapting domain specific natural language understanding. It solves the problem of simple natural language understanding and allows users to interact with machines using natural language. This work shall be of immense importance to the students of English Grammar who sometime feel harassed while cramming rules of counting and tagging the narrations. This program shall enable them to check their counting and tagging and also print the desired result at the click of the mouse. This software has the advantage of being user friendly and occupies limited space and also it's a GUI based.

8. CONCLUSION

Secondary data can be a valuable source of information for gaining knowledge and insight into a broad range of issues and phenomena. Review and analysis of secondary data provides a less-effective way of addressing narratives and also helps in determining the direction of ADA. It complements, but does not replace primary data collection and should be the starting place for any research. Introduction of probabilistic work can lead us to deal with voluminous and enlarged situations which have become the requirement of the day. In

future this concept of probabilistic inclusion can give us inference about many untouched results.

9. ACKNOWLEDGEMENTS

Authors are grateful to the reviewer for suggesting modifications in this paper.

10. REFERENCES

[1] Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In Proc. of NAACL.

[2] Alansary S, Nagi M, Adly N. (2008). Towards Analyzing the International Corpus of Arabic (ICA). 8th International Conference on Language Engineering, Egypt.

[3] Atwell E, Al-Sulaiti L, Al-Osaimi S, Abu-Shawar B.(2004). A Review of Arabic Corpus Analysis Tools. Proceedings of JEP-TALN'04 Arabic Language Processing.

[4] Banko M, Moore R. C. (2004). Part of Speech Tagging in Context. Proc of the 20th international conference on Computational Linguistics, Switzerland.

[5] Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010. Summarizing micro logs automatically. In Proc. of NAACL.

[6] Brendan O'Connor, Michel Krieger, and David Ahn.2010b. TweetMotif: Exploratory search and topic summarization for Twitter. In Proc. of ICWSM (demo track).

[7] Diab M., Hacioglu K. and Jurafsky D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. proc. of HLTNAACL'04: 149–152.

[8] Emanuele Pianta and Christian Girardi and Roberto Zanoli. 2008. The TextPro tool suite. In Proceedings of LREC, Marrakech, Morocco.

[9] Finin Tim, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowd sourcing. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.

[10] Freeman A and khoja (2001). Brill's POS tagger and a morphology parser for English. In ACL'01 Workshop on Arabic language processing.

[11] Hendrickx Iris and Su Nam Kim and Zornitsa Kozareva and Preslav Nakov and Diarmuid ´O S´eaghdha and Sebastian Pad´o and Marco Pennacchiotti and Lorenza

[12] Romano and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. In Proceedings of the 5th SIGLEX Workshop on Semantic Evaluation, Uppsala, Sweden.

[13] Hulth, A. (2003). Improved automatic keyword counting given more linguistic knowledge. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2003, 216-223.

[14] John Lafferty, Andrew McCallum, and Fernando Pereira. (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of ICML.

[15] Jurafsky D., Martin J.H. (2008). Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing. IInd Edition.

[16] Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on Twitter from biased and noisy data. In Proc. of COLING.

[17] Maamouri M, Cieri C. (2002). Resources for English Natural Language Processing at the LDC. Proceedings of the International Symposium on the Processing of English ,Tunisia, pp.125-146.

[18] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19:313–330.

[19] Philips.Lawrence 1990. Hanging on the Metaphone. Computer Language, 7(12).

[20] Plas, L. van der, Pallotta, V., Rajman, M., & Ghorbel, H. (2004). Automatic keyword extraction from spoken text. A comparison of two lexical resources: the EDR and WordNet. In Lino, M.T., Xavier,

[21] Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In Proc. of WIIAT.

[22] Suzuki, Y., Fukumoto, F., & Sekiguchi, Y. (1998). Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles. SIGIR, 1998, 373-374.

[23] Thelwall Mike, Kevan Buckley, and Georgios Paltoglou. 2011. Sentiment in Twitter events. Journal of the American Society for Information Science and Technology, 62(2):406–418.

[24] Tim Buckwalter. (2004). Buckwalter Arabic Morphological Analyzer, Version 2.0. LDC Catalog No. LDC2004L02, Linguistic Data Consortium, www.ldc.upenn.edu/Catalog.

[25] Tlili-Guiassa Y. (2006). Hybrid Method for Tagging Arabic Text. Journal of Computer Science 2 (3): 245-248.