Language detection and translation using n-gram and statistical machine translation approach

Smruti Tahasildar
BE IT Student,
K J Somaiya Institute of Engineering & IT, Sion, India.
smruti.t@somaiya.edu

Abstract With the increasingly widespread use of computers And the Internet in India, large amount of information in different languages are becoming available on the web. Information on the internet may seem limitless. Therefore translation becoming an urgent need in the Indian context .So in this paper we will present an idea that will help us to improve the information flow. Implementation of efficient language detection and translation application will solve this language barrier problem. Using this Language detection and translation application, we can identify the language of the text which provide information to potential readers and therefore improve the flow of ideas. This translator provides instant translations between different languages. With this translator, we hope to make information universally accessible and useful, regardless of the language in which it's written. In this paper, we discuss the N-gram approach and Statistical Machine Translation.

Keywords — ***Language, Detection, Translation, Machine, System, Algorithm, Model, N-gram, Statistical Machine Translation.***

I. INTRODUCTION

One of the most important advances of our time is experienced in the field of communication. The most important thing of communication is the language that is considered to remain away from these advances. The language problem is one of the most important problems to be solved on each passing day in the globalized world. Despite increasing in the amount of available documents, unfortunately, there is no opportunity to use the language of unknown documents. In order to make available sources of information more useful, language identification and language comprehension have significant duty. So, language identification is the first step of understanding the language.

As the web grows, language identification and its translation in general is becoming an important issue. Web environment provides enormous amount of documents, often in the other languages which is not known to the user, which makes the task of identification uneasy. N-gram based approach, which is the basic method for text categorization, could be used with slight modifications to perform language detection. The main purpose of the system is to identify the language of the text uploaded by the user and translate it to the user desired language. It will help the researchers, student, and teachers to study the topics thoroughly. Translation is the communication of the meaning of a source-language text by means of an equivalent target-language text.

In our study, we focus on the identification of document based language. Language identification approaches are divided into two methods: linguistic methods and statistical methods. Linguistic method which is approach in language identification estimates the language in the documents according to the grammar rules belonging to languages. The document based approach in language identification, one of the linguistic method estimates the language according to the rules of grammar in language documentation. It makes searching according to the frequency of searches for words in the document and makes scoring them. The availability of the automated translation system makes it possible to translate an entire corpus into a new language. This paper shows that it is effectively feasible if input text is having maximum characters. The quality of the translation depends on the amount of manually transcribed data used for training the automatic machine translation(MT) component.

In this work, we propose a statistical machine translation model that automatically converts the original form of text to the desired language form, which in turn are used for building more efficient translation application.

II. RELATED WORK

To express language statistical, the order of letters, the presence of certain keywords, frequencies of short words (a combination of presence) is decisive and each language is represented its descriptive features .For it is extracted from the languages features and for this n-gram feature extraction method is used. Using these features in the form of a feature

vector representation of documents is one of the most widely used methods. The purpose of this study, to date, comparison of accuracy rate of the language identification approach based on classification methods, and so, reveal the most appropriate methods. Preprocessing and extracting features processes are the first step to identify languages.

III. APPLICATION FLOW

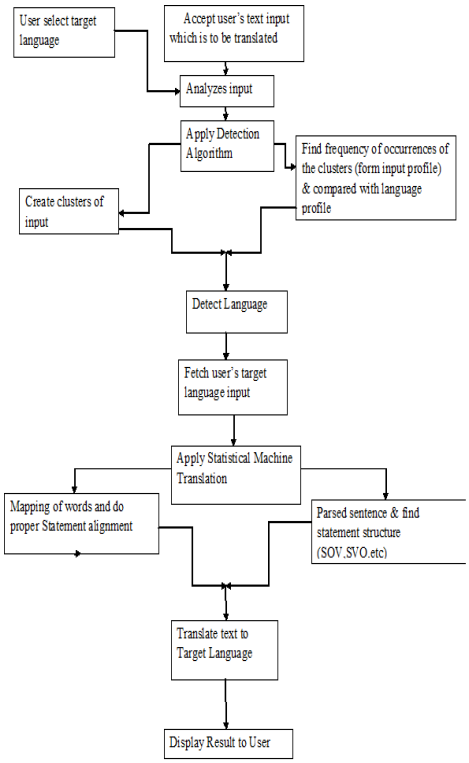

Figure 1. Flow of system

Module 1: Inputting the text document to the Application
User gives the text document as a input to the Application & choose the target language in which he wants to convert.

Module 2: Analysis of Text Document
Application will process the document & it will calculate the language profile needed to identify the language.

Module 3: Identification of Languages
The language profile calculated by the Module 2 is used to identify the language of the text.
For this, we will use the N-Gramm Approach.

Module 4: Translation of the Text document.
It will translate the text document to the desired language as per the user's input given to the system.

IV. METHODOLOGY, TECHNIQUES AND ALGORITHMS

4.1 N-Grams Approach

An N-gram is an N-character slice of a longer string. Although in the literature the term can include the notion of any co-occurring set of characters in a string (e.g., an N gram made up of the first and third character of a word), in this paper we use the term for contiguous slices only.

Typically, one slices the string into a set of overlapping N-grams. In our system, we use N-grams of several different lengths simultaneously. We also append blanks to the beginning and ending of the string in order to help with matching beginning-of-word and ending-of-word situations. (We will use the underscore character ("_") to represent blanks.) Thus, the word "TEXT" would be composed of the following N-grams:

bi-grams: _T, TE, EX, XT, T_
tri-grams: _TE, TEX, EXT, XT_, T_ _
quad-grams: _TEX, TEXT, EXT_, XT_ _, T_ _ _

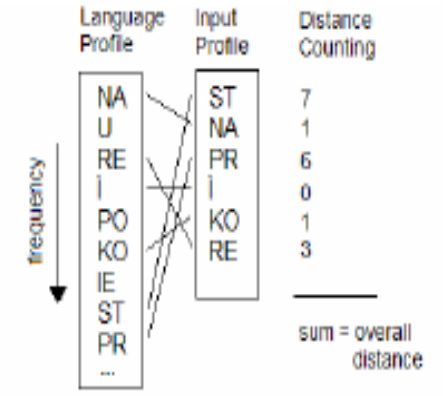

Fig 2. Comparison between unknown text and one of languages

In general, a string of length *k*, padded with blanks, will have *k*+1 bi-grams, *k*+1tri-grams, *k*+1 quad-grams, and so on. The key benefit that N-gram-based matching provides derives from its very nature: since every string is decomposed into small parts, any errors that are presented to affect only a limited number of those parts, leaving the remainder intact. If we count N-grams that are common to two strings, we get a measure of their similarity that is resistant to a wide variety of textual errors.

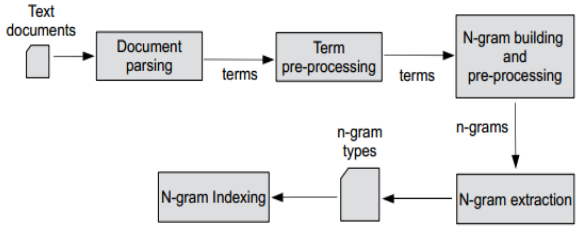

Figure 3.A common architecture of an n-gram extraction framework

In Figure 1, we see a common architecture of an n-gram extraction framework. This framework usually includes:

1) Document parsing – it parses terms from input documents.

2) Term pre-processing – in this phase, various techniques like stemming and stop-list are applied for the reduction of terms.

3) N-gram building and pre-processing – it creates an ngram as a sequence of n terms. Sometimes, n-grams are not shared by text units (sentences or paragraphs). It means, the last term of a sentence is the last term of an n-gram and the next n-gram begins by the first term of the next sentence.

4) N-gram extraction – the main goal of this phase is to remove duplicate n-grams. The result of this phase is a collection of n-gram types with the frequency enclosed to each type.For example, n-gram types with a low frequency are removed. Evidently, it is not appropriate to apply this post-processing in any application. It can be used only when we do not need these n-gram types which is not our case.

5) N-gram indexing – a common part of such a framework is n-gram indexing. A data structure is applied to speedup access to the tuple ngram, id, frequency, where ngram is a key; it means the ngram is an input of the query and id and frequency form the output. This solution is usable for Boolean or Vector models. Although, it is necessary to create other data structures for specific document and query models, we must always consider this global storage of the tuples for the whole document collection. Existing techniques of the n-gram extraction suppose only external sorting algorithms. These methods must handle a high number of duplicate n-grams which are removed after the frequency is computed. It results in high time and space overhead. In this article, we show a time and space efficient method for the n-gram extraction; we do not consider various methods of document parsing, term pre-processing, and n-gram building and pre-processing. We show that we can use data structures well known in the area of database management systems and physical database design for this purpose. In this way, we utilize the same data structures for the n-gram indexing and the n-gram extraction. Additionally, we show a high scalability of our method; it is usable for large document collections including up-to 109 n-grams regardless the amount of the main memory.

4.2 STATISTICAL MACHINE TRANSLATION

Statistical MT models take the view that every sentence in the target language is a translation of the source language sentence with some probability. The best translation, of course, is the sentence that has the highest probability. The key problems in statistical MT are: estimating the probability of a translation, and efficiently finding the sentence with the highest probability.

We suppose that the sentence *f* to be translated was initially conceived in language *E* as some sentence *e*. During communication *e* was corrupted by the channel to *f*. Now, we assume that each sentence in *E* is a translation of *f* with some probability, and the sentence that we choose as the translation (ˆ*e*) is the one that has the highest probability. In mathematical terms [Brown et al.,1990],

$$\hat{e} = \arg\max_{e} P(e|f)$$

Intuitively, $P(e|f)$ should depend on two factors:

1. The kind of sentences that are likely in the language *E*. This is known as thelanguage model — P(*e*).
2. The way sentences in *E* get converted to sentences in *F*. This is called the translation model — P(f|*e*).

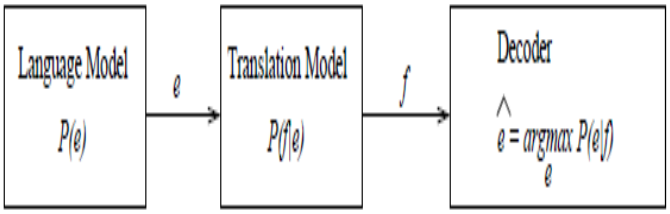

Figure 4. The Noisy Channel Model for Machine Translation

$$\hat{e} = \arg\max_{e} \frac{P(e)P(f|e)}{P(f)}$$

Since *f* is fixed, we omit it from the maximization & we get following equation,

$$\hat{e} = \arg\max_{e} P(e)P(f|e)$$

Thus, statistical translation requires three key components:

1. Language model
2. Translation model
3. Search algorithm

V. CONCLUSION

This paper introduced a translation approach to language modelling. Specifically we used n-gram approach model to detect an input text and using statistical machine translation input text is translated into desired language.

The tools and methodologies which are developed are can be used to develop a translation system that translates English to other morphologically rich languages. In future, advancement of this system lies in integrating with speech recognition systems.

REFERENCES

[1] Cavnar, W. B. and J. M. Trenkle : N-Gram Based Text Categorization. In: Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, UNLV Publications/Reprographics (1994), 161-175.

[2] Dunning, T.: Statistical Identification of Language. In: Technical report CRL MCCS-94--273, Computing Research Lab, New Mexico State University (1994).

[3] N-Gram Based Statistics Aimed at Language Identification, Tomáš ÖLVECKÝ Slovak University of Technology Faculty of Informatics and Information Technologies Ilkovičova 3, 842 16 Bratislava, Slovak Republic olvecky@stonline.sk

[4] Statistical Machine Translation ADAM LOPEZ University of Edinburgh

[5] Statistical Machine Translation Ph.D. Seminar Report by Ananthakrishnan Ramanathan

[6] Use of statistical N-gram models in natural language generation for machine translation,Liu, F.-H. ; Liang Gu ;

[7] Yuqing Gao ; Picheny, Michael Publication Year: 2003 IEEE CONFERENCE PUBLICATIONS

[8] Index-Based N-gram Extraction from Large Document Collections Michal Kratk ´ y, Radim Ba ´ ca, David Bedn ˇ a´ˇr, Jiˇr´ı Walder, Jiˇr´ı Dvorsky, Peter Chovanec

[9] Z. Ceˇ ska, I. Han ˇ ak, and R. Tesa ´ ˇr, "Teraman: A tool for n-gram extraction from large datasets," in Intelligent Computer Communication and Processing, 2007 IEEE International Conference on, 2007, pp. 209– 216.

[10] J. Pomikalek and P. Rychl ´ y, "Detecting Co-Derivative Documents in ´ Large Text Collections," in Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, Morocco. European Language Resources Association (ELRA), 2008, pp. 132–135.

[11] P. Xu, D. Karakos, and S. Khudanpur, "Self-supervised discriminative training of statistical language models," in Proc.IEEE ASRU, 2009.

[12] H. Printz and P. Olsen, "Theory and practice of acoustic con-fusability," Computer Speech and Language, vol. 16, no. 1, pp. 131–164, 2002.

[13] J.-Y. Chen, P.A. Olsen, and J.R. Hershey, "Word confusability measuring Hidden Markov Model similarity," in Proc. Interspeech, 2007.

[14] Q.F. Tan, K. Audhkhasi, P. Georgiou, E. Ettelaie, and S. Narayanan, "Automatic speech recognition systemchannel modeling," in Proc. Interspeech, 2010.

[15] Li, Wang, S. Khudanpur, and J. Eisner, "Unsupervised discriminative language model training for machine translation using simulated confusion sets," in Proc. Coling, 2010.

[16] A. Stolcke, "Entropy-based pruning of backoff language mod-els," in Proceedings ofthe DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, 1998.

[17] Ying Zhang, Almut Silja Hildebrand, and Stephan Vogel, "Distributed language modeling for n-best list re-ranking," in Proceedings ofthe 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, July 2006, Association for Computational Linguistics.

[18] M. Elhadad and J. Robin, "An overview of SURGE: a reusable comprehensive syntactic realization component," Department of Mathematics and Computer Science, Tech. Rep., 96-03,1996.

[19] H. Inozuka, M. Hosoi, Y. Aramomi, and K. Murata, "Sentence generation system by IPAL (SURFACE/DEEP)," IPSJ Technical Report.NL., vol. 99, no. 22, pp. 113–119, 1999.

[20] J.-H. Hong, S. Lim, and S.-B. Cho, "Autonomous language development using dialogue-act templates and genetic programming," IEEE Trans. Evolutionary Computation, vol. 11, no. 2, pp. 213–225, 2007.

[21] G. Ferguson and J. F. Allen, "TRIPS : An integrated intelligent problem-solving assistant," Proc. the 16th National Conference on Artificial Intelligence (AAAI-98), 1998.

[22] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington, "JUPlTER: a telephone-based conversational interface for weather information," Speech and Audio Processing, IEEE Transactions on, vol. 8, no. 1, pp. 85 –96, Jan. 2000.

[23] Y. Yagi, S. Takada, K. Hirose, and N. Minematsu, "Realization of concept-to-speech conversion for reply speech generation in a spoken dialogue system of road guidance and its evalution(speech processing)," IPSJ, vol. 48, no. 9, pp. 3300–3308, 2007.

