HANDLING AVIATION OOV WORDS FOR MACHINE TRANSLATION AND CORPUS CREATION

Saptarshi Paul
Assistant Professor, Computer Science Department,
Assam University Silchar, Silchar, Assam-788011, India
paulsaptarshi@yahoo.co.in

Bipul shyam Purkhyastha
Professor, Computer Science Department,
Assam University Silchar, Silchar,Assam-788011, India
bipul_sh@hotmail.com

Abstract - The Indian aviation industry has been one of the fastest growing in the world since the last decade. Aviation experts anticipate it to be the largest in the world by 2030. The accelerated growth in passenger number has been influential enough to encourage more job opportunities in the industry. The anticipated exponential growth also envisions increase in the usage of aviation OOV words and phrases. For communicating the various proceedings and news through the mainstream and social media, these aviation words have become more frequent nowadays. The aviation domain is heavily dependent on the use of OOV words, which, in turn, are structured words molded to fit the aviation domain. This causes a huge problem during the conversion from English to other languages using machine translation (MT). This paper explores the problems encountered during translation of such sentences and suggests how these sentences can be pre-processed before being fed into the MT software for better translations. This paper also investigates how these structured and OOV words can be handled in the English language itself, before being converted to any other language and surveys the prospects of maneuvering this tool to create a bilingual corpus for Machine Translation applications.

Keywords: Aviation, OOV words, Machine Translation, Corpus

1. Introduction

The need for translation of English to Indian languages has fueled the development of many MT systems in various domains such as medical, tourism etc. But till now translation of aviation domain has not been attempted for any Indian languages. The aviation market, being one of the fastest growing in India is creating huge job opportunities and the need for translation of aviation sentences and terminologies into Indian languages are becoming stronger with time. The aviation domain is heavily dependent on OOV words and structured English sentences, which are unique to only aviation. These words and phrases make it harder to translate. With airliners using social media (Twitter, Facebook, etc) as a vital means of advertisement and communication these words related to aviation are

finding its way in regular day to day reads such as newspapers. The need of the hour is, therefore, to have a tool that can identify these structured words, phrases and OOV words and replace them with the corresponding meaningful word in English so that MT systems can easily translate them. The MT systems in turn have to be trained using aviation parallel corpus (English and target language) for proper translation. The absence of such corpus is also a void that needs to be filled up. A tool that can address both the direct feed to MT tools and help in creation of a parallel corpus is much needed.

2. Related work

The only successful existing systems for translation from source to target languages for aviation domain is mainly restricted to French to English TUAM AVIATION (1976) [Isabelle and Bourbeau,(1985)][Paul and Purkaystha, (2018)]. Other systems (similar but not directly related to aviation) include approaches where an approach based on feedback of the MT software has been proposed , [Waibel *et al,* (2008)]which works for morphologically richer source language and where they have tried to extract unknown words and later found their meanings and fed them to the MT translator. Also works is included by [Yves and Etienne,(2005)] where they used analogical learning. [Gupta and Lehal, (2011)] suggests preprocessing for indian languages such as Punjabi which has been attempted. The impact of preprocessing has been elaborately described by [Gunal and Uysal, (2014)].

3. The Problem

Aviation being a highly technical and specialized domain uses OOV words in huge numbers. The main challenge is to identify this OOV words and transfer them to their corresponding meaningful forms before being translated to the target language [Paul and Purkhyastha, (2019)]. Also another challenge is to identify the structured words used in aviation and social media, to identify them and then replace them with corresponding English words before going for the translation. Along with this OOV and structured words are special characters like @, #, etc which are impossible for existing translators to identify, understand and convert. Let us take an example:

Fig 1. A Tweet by Airport Authority of India

If we are to copy the contents of the above tweet directly and use any standard converter like “Google Translate” to convert it, then it will be unable to translate the tweet properly. The translator is unable to remove the special characters and symbols like “#” and “@” and also unable to indentify OOV words like “MoCA” and “UDAN” which means “Ministry of Civil Aviation” and “Ude Desh Ka Aaam Nagrik” respectively.

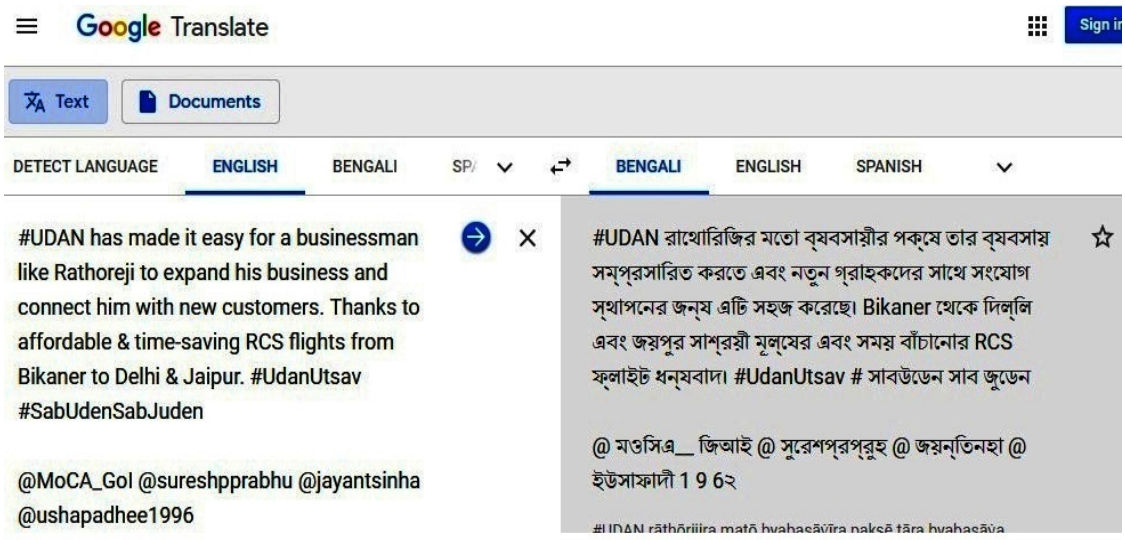

Fig 2. Google Translate unable to translate (English-Bengali) tweet contents with special characters and aviation OOV words and structured words.

Likewise translators are unable to translate OOV words and structured words which are heavily used in aviation.

Table 1. Examples of aviation OOV words

OOV words	Meaning
BLR	Bangalore Airport
TFC	Traffic
FL	Flight Level
DSND	Descend
SBOUND	South Bound
FLT	Flight
MAA	Chennai Airport
6E	Indigo Airlines
AI	Air India
AAI	Airport Authority Of India
ILS	Instrumental Landing System
DVOR	Doppler Very High Frequency Omni Range

Similarly all the airports names (IATA and ICAO call signs), aircrafts, instruments used in aviation have specific call-signs and names (OOV words) and so does all the routes through which aircrafts traverse. These are used symmetrically all over the world. These structured and OOV words are regularly used in social media and in mainstream media to complicate the regular sentences more than ever. Structured English is a very unconventional language that is usually spoken, written and used especially by a particular group of people (in our case by people related to aviation). They generally refer to particular set of words and meanings but can include longer-expressions and idioms.

4. The Problem: An Example

While dealing with numerous aviation posts and conversations, the MT software cannot understand what they actually mean. Ultimately the MT software gives an output that is intelligible or poorly translated with the structured words and OOV words un-translated and at best transliterated. The most common problem faced by the MT software is that they cannot understand the sentences (source language, here English) that contain such structured words and OOV words as they are generally not part of the training database. For example: A conversation reads: "*Sbound will be in risk. So we are heading towards MAA*". This means "Proceeding towards south will be in risk. So we are heading towards CHENNAI AIRPORT". If we are to translate the above sentence using a standard MT software (Microsoft Bing-translate/Google Translate) we see that it does not understand "Sbound" and so does not translate/transliterate it and considers "MAA" (aviation OOV word) as the Bengali equivalent of mother and translates it to "MA" in Bengali, which is of course a wrong translation and reads as: **"**Sbound **ঝুঁকির মধ্যে পরবে । তাই আমরা মা দিকে এগিয়ে যাচ্ছি ।"**

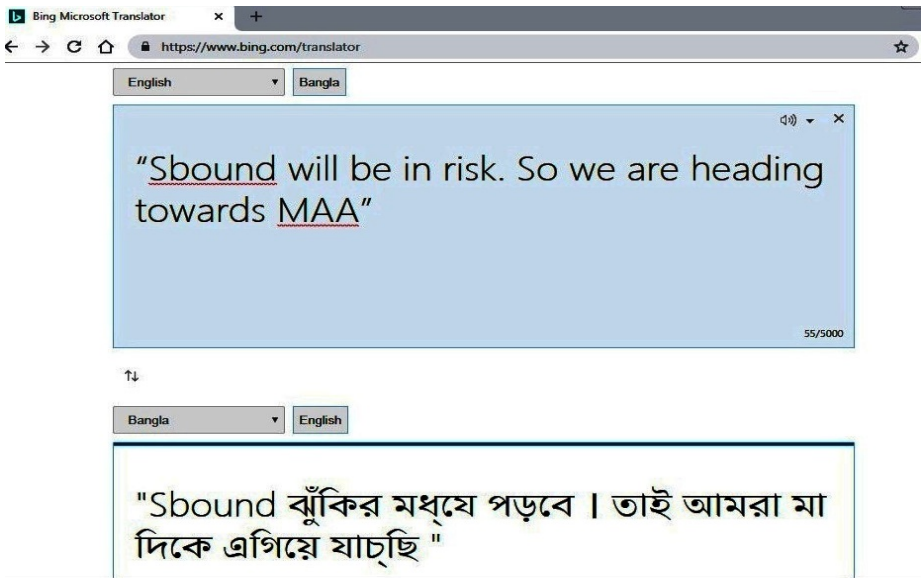

Fig 3. Inability of Microsoft Bing Translator to translate sentences containing aviation OOV words.

5. The Proposed Solution

A simple solution is to device a tool where such sentences containing OOV words, special characters and structured words can be transformed into regular English sentences, before being fed into the translation software, and thus can be understood by the MT software and in turn, be correctly translated. This tool can also be used to create parallel Bilingual (Source-Target) databases through which MT systems can be trained to build models capable of handling OOV words and such. This can be done by creating a database of such structured words and OOV words along with their meanings. This database can be referenced and whenever such words are encountered it can be replaced with the corresponding meaningful word in the sentence. This preprocessed sentence can then be fed to the MT software for successful and fruitful transformation.

5.1 Methodology

Using Python Programming we aim to achieve the following:

1) Replace the OOV words used in aviation with their everyday in use general English equivalent/meanings.

2) To identify and remove special characters such as @,_,# etc.The structured and OOV words are stored in a database that is referred by the program.

5.2 Working of the program

Using Python the program is written so as to replace the structured and OOV words used in aviation with the general English/MT understandable words so that the MT translators can easily understand them. A database of such structured and OOV words along with their meanings is created. It is an open database where the process of adding new words is a continuous process. In the database the words are separated from their meanings by the delimiter '='. The '**replace**' command replaces common in-short terms with their full terms. Moreover the program also uses replace command to replace "#" (hashtags), "@" (at-the-rate), "_" (under-score) and other special characters. Taking the previous example the statement posted in social media by an aviation expert was "*Sbound will be in risk. So we are heading towards MAA*". Our program replaces it as "*Proceeding towards south* will be in risk. So we are heading towards *Chennai airport*' where the OOV term 'Sbound' in aviation means *going towards south* and "MAA" means *Chennai Airport* and it is as stored in the database. The "filename" gives the name of the file where terms are stored. We have used the "**accessmode**" as **'read'** operation. The program takes the input data (containing hashtags, under-score etc) using the "**input()**" function. The "**strip()**" function eliminates the *hastags(#*).The "**replace()**" function replaces the structured and OOV words with the equivalent machine translatable meaningful words. The '**translator**' library in the python program is added so as to translate the terms to their common meanings. The "**split()**" function splits the string into tokens. The Solution has been provided in the program through a simple approach, we have taken the help of manually prepared data database, to detect type of noise (special characters, OOV words) and replace them with corresponding meaningful words that can be easily translated by MT software.

6. Results and Output

6.1 *As a preprocessing tool for direct input in MT Software:*

Continuing with the previous example as shown in fig.3, we feed the same sentence to our proposed software and the output is: "Proceeding towards south will be in risk. So we are heading towards Chennai airport"

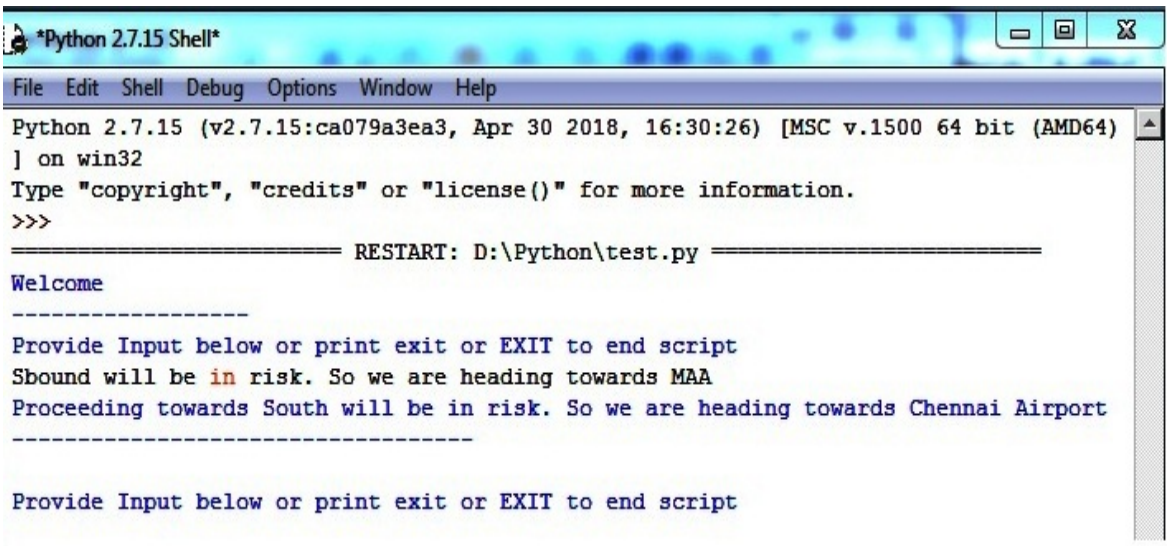

Fig 4. Input and output of our software for the said example

Now, on providing the above output to Microsoft Bing, the translation is as follows:

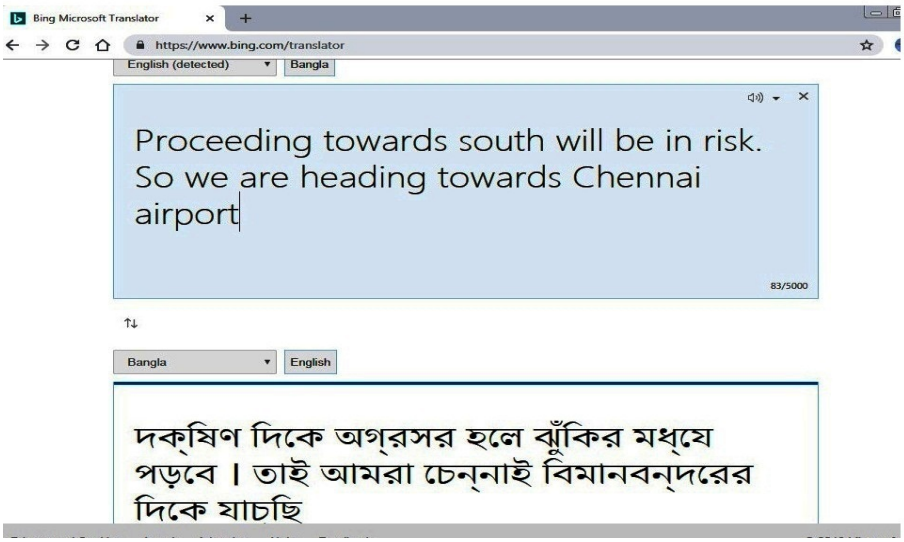

Fig 5. Proper translation by Microsoft Bing Translator on input from our proposed s/w.

Here we see that Microsoft Bing Translate has been able to translate the sentence correctly, from English to Bengali keeping the meaning intact which is:

"**দক্ষিণের দিকে অগ্রসর হলে ঝুঁকির মধ্যে পরবো। তাই আমরা চেন্নাই বিমানবন্দরের দিকে যাচ্ছি।**"

This tool can and is being used as a preprocessing tool to get correct results from commercially available MT systems, but one of the capabilities of this tool lies in its ability to assist in the creation of English (Source) to target language bilingual corpus for the aviation domain.

6.2. As a tool for creating Bilingual Corpus:

The tool can be used to assist in the creating of any bilingual corpus (English- target language). Currently it is being used to create English to Bengali Corpus for the aviation domain. The aviation data / sentences collected from various sources are being directly fed into the tool to get regular English sentences which are then placed as the source language and then the appropriate Bengali sentences (target language) are created and placed in the corresponding column. This application can be extended to any other Target language .The snapshot of the corpus is provided in Fig 6 to give an idea of the application of the software as a corpus-creation assistance tool. This Bilingual Corpus can in turn be used to train any SMT / NMT system.

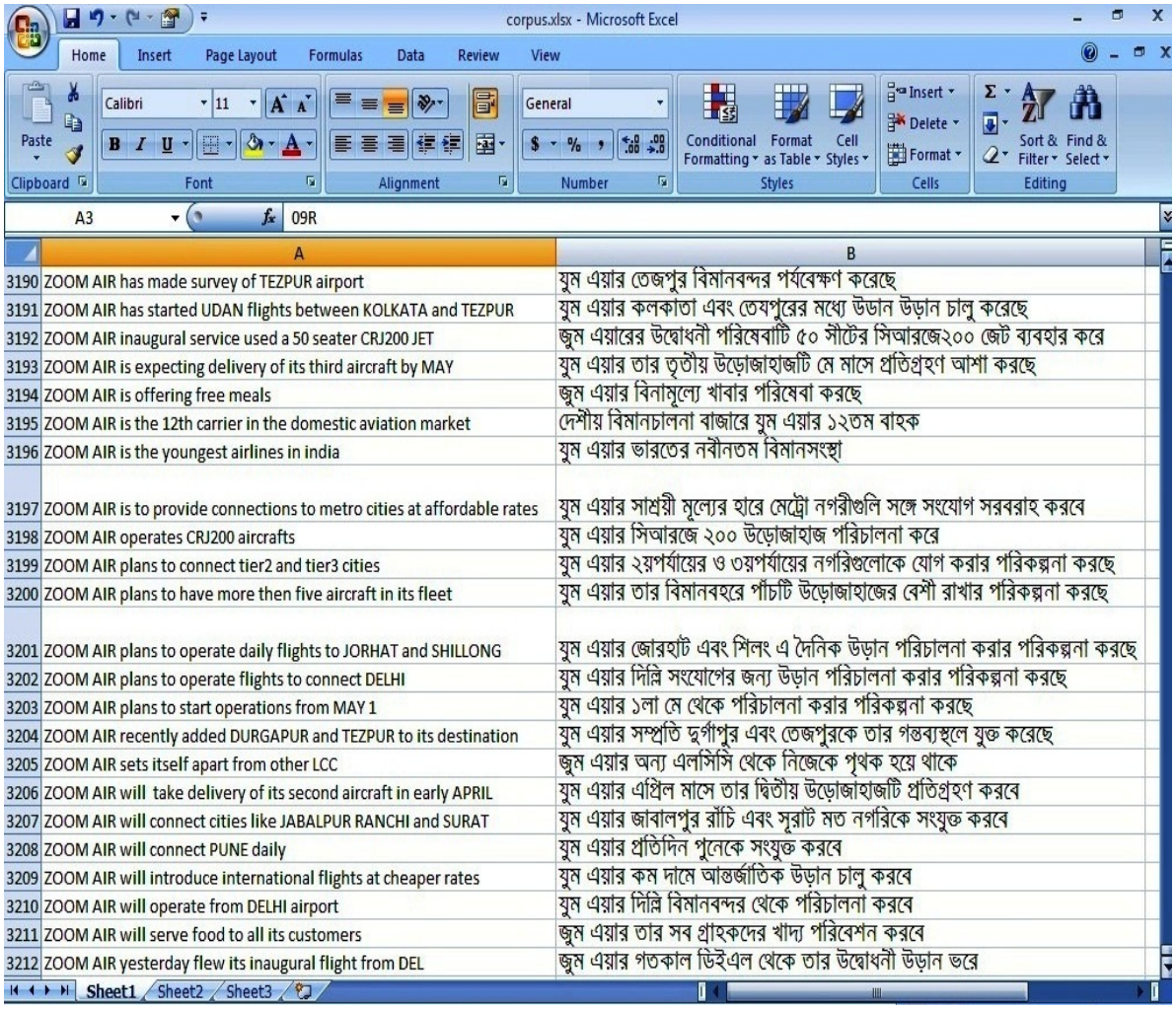

Fig 6. Bilingual aviation Corpus formed with help of the preprocessing tool

7. OOV Sentences Sources

Gathering aviation related structured and OOV words and their meaningful translations in English language from dependable/genuine sources is a tough work. We have referred official publications from:

1. Airport Authority Of India (AAI) - Manual of Air traffic Services [AAI-Reports, (2020)]

2. Directorate General of Civil Aviation, (DGCA) manuals and accident/incident reports [DGCA-reports, (2020)]

3. International Civil Aviation Organization (ICAO) reports on safety/incidents. [ICAO Air Safety Reports, (2018)]

4. National Transportation safety board-Aviation accident data summery [NTSB aviation database, (2018)]

Also data from NASA-ASRS [ASRS Report Sets, (2019)] incident reports, aviation newsletters and aviation related blogs has been included.

8. Conclusion

Preprocessing source language sentences (for technical domain such as aviation) to normalize them before being fed into MT system for direct translations is a logical step. Till now, there has been no preprocessing software for a specialized domain such as aviation and aero-space. The above software can be used to assist in the creation of English (Source) to any target language parallel bilingual corpus. It is not only a pre-processing tool for corpus creation but also a standalone tool that can be used to normalize English aviation sentences before being fed into any standard off-the-shelf translation software and get better translation quality.

9. Future Works

The monolingual database used in the tool is an open one and as we come across new OOV words, we can keep on updating it. To give the software a better interface and the ability to link with standard MT systems are planned for the future.

References

[1] Gunal,S; Uysal,A.K;(2014) The impact of preprocessing on text classification, Information Processing & Management 50(1) January 2014, pp:104 - 112
[2] Gupta,V; Lehal, G. S;(2011) Processing Phase of Punjabi Language Text Summarization, Information Systems for Indian Languages, ICISIL 2011, pp 250-253
[3] Isabelle,P; Bourbeau,L;(1985), "Tuam Aviation: its technical features and some experimental results" , Computational linguistics, volume 11, number 1, January-March 1985, pp:18-27
[4] Paul, S; Purkaystha,B.S; (2018), "NLP Tools used in civil aviation: A survey", International Journal of Advanced Research in Computer Science , Volume 9, No. 2, March-April 2018 pp: 109 -114
[5] Paul,S; Purkhyasta.B.S; (2019) English to Bengali Transliteration tool for OOV words common in Indian civil aviation, journal of advanced database management and systems, 2019, 6(1), pp 23-32
[6] Waibel,A; Eck,M; Vogel,S; (2008), "Communicating Unknown Words in Machine Translation" , Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco
[7] Yves,L; Etienne,D;(2005) "Purest ever example-based machine translation: Detailed presentation and assessment", Machine Translation , December 2005, volume 19, Issue 3–4, pp 251–282
[8] AAI-Reports. Airport authority of India, www.aai.aero/hi/system/files/resources. Accessed 3 Jan. 2020.
[9] DGCA-reports, dgca.gov.in. Accessed 10 Aug. 2019.
[10] ICAO Air Safety Reports. ICAO, www.icao.int/safety/airnavigation/AIG/Pages/default.aspx. Accessed 19 Mar. 2018.
[11] NTSB aviation database. www.ntsb.gov. Accessed 6 Mar. 2018.
[12] ASRS Report Sets - Aviation Safety Reporting System. NASA, asrs.arc.nasa.gov/search/reportsets.html. Accessed 25 Apr. 2019.