Understandability of machine-translated Hindi tweets before and after post-editing: perspectives for a recommender system

Comprensibilidad de tweets en Hindi traducidos por un sistema de traducción automática antes y después de post-edición: perspectivas para un sistema de recomendación

Ritesh Shah
Université Grenoble-Alpes,
GETALP-LIG,
Grenoble, France
ritesh.shah@imag.fr

Christian Boitet
Université Grenoble-Alpes,
GETALP-LIG,
Grenoble, France
christian.boitet@imag.fr

Resumen: En el proceso de construcción de un sistema de recomendación basado en tweets en Hindi para un proyecto, queremos determinar si los resultados brutos de traducción automática (TA) podrían ser útiles. Hemos recogido 100K tales tweets y experimentado con 200 de ellos como paso preliminar. Por lo menos el 50% de los tweets traducidos por TA resultaban comprensibles para hablantes de inglés, mientras que por lo menos el 80% de comprensibilidad sería necesario para que la TA fuera útil en este contexto. Posteriormente, hemos post-editado los resultados de TA y hemos observado que la comprensibilidad aumentó al 70%, mientras que el tiempo de post-edición fue 5 veces menor que el tiempo de traducción humana. Esbozamos, así, un escenario para producir un sistema de TA especializado para traducir (automáticamente) tweets de hindi a inglés, consiguiendose que del 70% al 80% de los tweets obtenidos fueran comprensibles.
Palabras clave: tweets en Hindi, sistemas especializados de traducción automática, post-edición, comprensibilidad, sistema de recomendación

Abstract: In the process of building a recommender system based on Hindi tweets for a project, we want to determine whether raw Machine Translation (MT) results could be useful. We collected 100K such tweets and experimented on 200 of them as a preliminary step. Less than 50% of the machine-translated tweets were understandable by English speakers, while at least 80% understandability seems to be required for MT to be included as a useful feature in this context. We then post-edited the MT results and observed that understandability reached 70%, while post-editing time was 5 times less than human translation time. We outline a scenario to produce a specialised MT system that would be able to translate (fully automatically) 70% to 80% of the tweets in Hindi into understandable English.
Keywords: Hindi tweets, specialised MT system, understandability, recommender system, post-editing

1 Introduction and objectives

The operational architecture of a Machine Translation (MT) system is determined by precise conditions of the use and development of the system. For instance, the architecture changes depending on the role of MT system users (say, authors, professional translators), the language pairs involved, or when availability of resources is a primary constraint. When the task is simply to help people understand an unknown or little known language, the design of the MT system is driven by coverage and automaticity rather than by the output quality, while only the gist of a translation is to be conveyed (Boitet et al., 2009).

An interesting case is that of multilingual rec-

ommender systems relying on information mined from tweets in regional languages. The user of the system, for instance a tourist, might like to have a look at the top five translated tweets having influenced the recommendation (summarized as usual by 0 to 5 "stars"). A tweet translation system providing an operational quality output could be sufficient in such cases.

Keeping in mind the above context, we make a preliminary study of the understandability of tweet translations from Hindi to English, before and after post-editing them. For that, we randomly selected 200 tweets from the 100K collected, had them translated by Google Translate (GT), evaluated their understandability *as is* by English (non-Hindi) speakers, and asked a few Hindi speaking colleagues to post-edit the MT results (which we call "pre-translations") using the iMAG/SECtra (Huynh, Boitet, and Blanchon, 2008) web tool, giving them simple post-editing (PE) guidelines. In particular, they were asked to do minimal editing and not to aim at "normalizing", improving, or inserting missing information, and to write down the total time it took them to post-edit each tweet.

We then asked the same English (non-Hindi) speakers to evaluate again the proportion of understandable tweets. That rate rised from less than 50% before post-editing to more than 70% after post-editing. In the context of a recommender system and of the scenario sketched above, if more than 20% (or perhaps 30%) of the (translated) tweets are ununderstandable, the usage value of the MT system would be null, because users would simply stop looking at the tweets. On the other hand, if only 1 out of 5 tweets is ununderstandable, they would continue to look at them when they are curious about the reason for a particularly good or bad recommendation, so that the *usage quality* of the MT system might be judged *good enough* or *useful* or only *usable*. Our real distinction is whether the MT results would be *used*, even sparingly, or not at all.

While the value of the *minimal rate of understandability* certainly depends on each person, we could not yet set up an experiment with many tweet users, as we wished. In fact, the above value of about 70% has been obtained by asking only 2 *English-only readers*.

In the following section, we elaborate on the data collection and preprocessing. Section 3 explains the experimental setup and procedure. Experimental observations and a scenario for building a good enough specialized MT system follow in the last two sections.

2 Dataset: Hindi Tweets

2.1 Technology and Twitter API constraints

There are numerous services presently available for providing customised social content data, including tweets (GNIP, 2015). For our tweet dataset, we make use of the Twitter search API to extract tweets. The search API (non-Streaming API) from Twitter allows the developer to obtain a maximum of 1.73M tweets/day through the *Application-user authentication* (AuA) and a maximum of 4.32M tweets/day through the *Application-only authentication* (AoA).

The search API returns a collection of tweets corresponding to the requested query and the specified query filters. As we want to investigate tweet translations from Hindi to English, and make use of the search API under the AuA mode with a query containing the language filter 'lang:hi'. The query allows us to extract Hindi (translation source language) tweets within the rate-limit specified by the API. We used an interactive Python programming environment for data preprocessing and development to collect 100K tweets in Hindi.

2.2 Preprocessing

The preprocessing of our data involved format-conversion of the tweet dataset into HTML files as required by the iMAG[1] framework. We also had to normalize a subset of characters (in particular, emojis) to avoid potential systemic problems on account of data encoding and decoding.

2.2.1 Data format

The extracted tweets are in the JSON[2] format that contains the metadata and the textual content of each tweet. We kept only the textual content ('text' field) and the tweet identifier

[1] interactive Multilingual Access Gateway
[2] JavaScript Object Notation

('id_str' field) of each tweet. We finally converted the messages to a set of HTML files, each containing a table of 100 rows and 3 columns as shown in Figure 1. A third column with 'enum' field is added programmatically during conversion for enumeration.

Sr	tweetID	Tweet content
1	607618765354733568	RT @varanasilive: जीतने की खुशी में अपना सर बुलंद मत करो क्योंकि जीतने वाला भी अपना gold मैडल सर झुका के हांसिल करता है
2	607618764989808640	RT @dwivedijispeaks: " योग सदियों से चली आ रही भारतीय संस्कृति और सभ्यता का सूचक है, इसका विरोध करना अपनी मातृभूमि का विरोध करना है। #NaMo…
3	607618762297114624	@ShivshankarS @ArvindKejriwal @VictoryForNamo @DrGPradhan @ashok_kansala @Gravim71 @iamdsp1 जात स्वाभाव ना छूटे टांग उठाके मूते
4	607618762221580288	RT @varanasilive: %%EMOJI-0001f449आज का सुविचार~%%EMOJI-0001f44c नसीब का प्यार, और गरीब की दोस्ती, कभी धोखा नहीं देती..

Figure 1: Tweets in Hindi(Devanagari script) in an HTML table with fields:*enum,id_str and text*

2.2.2 Emoji issues

In order to verify data robustness and systemic consistency for further experiments, we set up an existing iMAG for a few files. During the process, we identified a problem that manifested in the form of emoji(s) and emoticons which are frequently used in tweet texts. The incorrect handling of the UTF-8 mapping scheme for those Unicode points that code these emojis caused the setup to fail.

Our solution was to normalise, during the preprocessing step, a range of such special occurrences. We identified and converted characters in the following Unicode point ranges (emojiList-1, 2015) (emojiList-2, 2015) in such a way that it should be possible to restore them at the end of the translation process.

For instance, the character '\U0001F44C' is converted to '%%EMOJI-0001f44c'. An example can be seen in row 4 of Figure 1.

3 Experiment

3.1 About iMAG/SECTra

iMAG/SECTra is a post-editing framework which internally employs GT by default (and any number of available MT servers) and allows integration of specialised MT systems. The system provides pre-translations to the post-editor and allows post-editing in various modes. It also allows post-editors to grade the quality of post-editions and record total time for post-editing (Tpe_{total}). In iMAG/SECTra, each segment has a *reliability level* [3] and a *quality score* between 0 and 20[4] (Wang and Boitet, 2013). While the reliability level is fixed by the tool, the quality score can be modified by the post-editor (initially, it is that defined in his profile) or by any reader.

The quality of the PE of a segment is deemed to be *good enough* if its quality score is higher or equal to 12/20.

3.2 Experimental setting

Our experimental procedure has two parts:

1. evaluating the understandability of pre-translations
2. post-editing pre-translations and estimating the output quality in relation with the post-editing times recorded by the post-editors.

First, we randomly selected two Hindi tweet datasets containing 100 tweets each (twTxtSet1 and twTxtSet2) and then we set up an iMAG/SECTra for post-editing the tweets.

3.2.1 Pre-translation understandability

In order to determine the proportion of understandable pre-translations (that is, tweets translated by GT), 2 participants speaking English and *no Hindi* were selected. Each participant was asked to give a score of 1 if a (translated) tweet was found to be *understandable* and 0 otherwise. The proportion of understandable tweets was recorded as **39%** for twTxtSet1 and **45%** for twTxtSet2.

[3] * for dictionary-based translation, ** for MT output, *** for PE by a bilingual contributor, *** for PE by a professional translator, and ***** for PE by a translator "certified" by the producer of the content.

[4] 10: pass, 12: good enough, 14: good, 16: very good, 18: exceptional, 20: perfect. 8-9: not satisfied with something in the PE. 6-7: sure to have produced a bad translation! 4-5: the PE corresponds to a text differing from that of the source segment. That happens when a sentence has been erroneously split into 2 segments and the order of words is different in the 2 languages. 2: the source segment was already in the target language.

3.2.2 Post-editing methodology

Speakers of both Hindi and English were selected to post-edit the pre-translations, thereby jotting down their Tpe_{total} for each tweet. To be in line with the envisaged scenario, where a tweet reader knowing both languages might conceivably (but rarely) correct a translation (*contribute* in Google's words), while no other reader would independently contribute on the same tweet, each tweet was post-edited only once. Monolingual and bilingual participants were then asked to score the post-edited pre-translations for understandability.

Even though post-editing was to be done in a minimal time possible, the post-editor was allowed to quickly label a tweet with a single word which would help human understanding and further elicit the context of certain tweets. This label was meant to be added but without taking much time. For instance, spam or derogatory tweets could be labeled as "((??spam??))", and code-mixed tweets could be labeled as "((??mixing??))". The label delimiters were pre-decided to separate them from the original tweet text. No set of labels was prepared beforehand. Labels were introduced by the post-editors themselves. The most frequent were {"news", "philosophy", "politics", "sports", "joke", "humour", "sarcasm", "quote"}

4 Observations

4.1 Post-editing statistics

Table 1 shows the total post-editing times (in mins) for twTxtSet1 and twTxtSet2. We observe and note additional statistics (given in the term definitions). More importantly, we obtain the quality measure which stands at **56.1%** for twTxtSet1 and **73.6%** for twTxtSet2.

Dataset	**#logical-pages**	**#segments**	**#source-words**	$Tpe_{total-mn}$
TwTxtSet1	17	331	1843	**162.6**
TwTxtSet2	18	356	1780	**93.7**

Table 1: Observed PE statistics

Dataset	**pages-std**	**mn per std_page**	$Thum_{estim-mn}$	**Quality**
TwTxtSet1	7.4	21.97	444	**56.1**%
TwTxtSet2	7.1	13.19	426	**73.6**%

Table 2: Calculated PE statistics

Term definitions used in Table 1 and 2:
std_page: consists of 250 words
#segments: number of translated segments
#source-words: number of words in source dataset
#logical-pages: number of PE pages in SECTra
pages-std: #source-words/250
$Tpe_{total-mn}$: Tpe_{total} in minutes
mn/std_page: $Tpe_{total-mn}$/pages-std
$Thum_{std\_page} = 60$
$Thum_{estim-mn}$: $Thum_{std\_page}$ * pages-std

In Table 2, the quality formula used (NII[5] lecture notes (Boitet et al., 2009)) is as follows (assuming that the human time to produce a translation draft for a standard page is 1 hour):

$$Q = 1 - 2/100 \times \frac{Tpe_{total-mn}}{Thum_{estim-mn}} \times Thum_{std\_page}$$

Examples:
$Q = 40\%$ if $Tpe_{total} = 30mn/p$ (8/20)
$Q = 60\%$ if $Tpe_{total} = 20mn/p$ (12/20)
$Q = 90\%$ if $Tpe_{total} = 5mn/p$ (18/20)

We proceed to add a few illustrative examples with descriptions to better visualise pertinent stages of our experimental procedure and observations.

4.1.1 Example 1: An ununderstandable MT output

H: मंजिल मिले ना मिले ये तो मुकदर की बात है !
T: These services are not met, then the floor is Mukdr!
PE: Destination whether met or are not met, then it is luck!

H: *A Hindi tweet as an input to iMAG*
T: *The machine-translated Hindi tweet (pre-translation)*
PE: *The post-edited output with score 14 (good) and with a Tpe_{total} of 33 seconds*

[5]National Institute of Informatics, Japan

4.1.2 Example 2: Post-editing environment

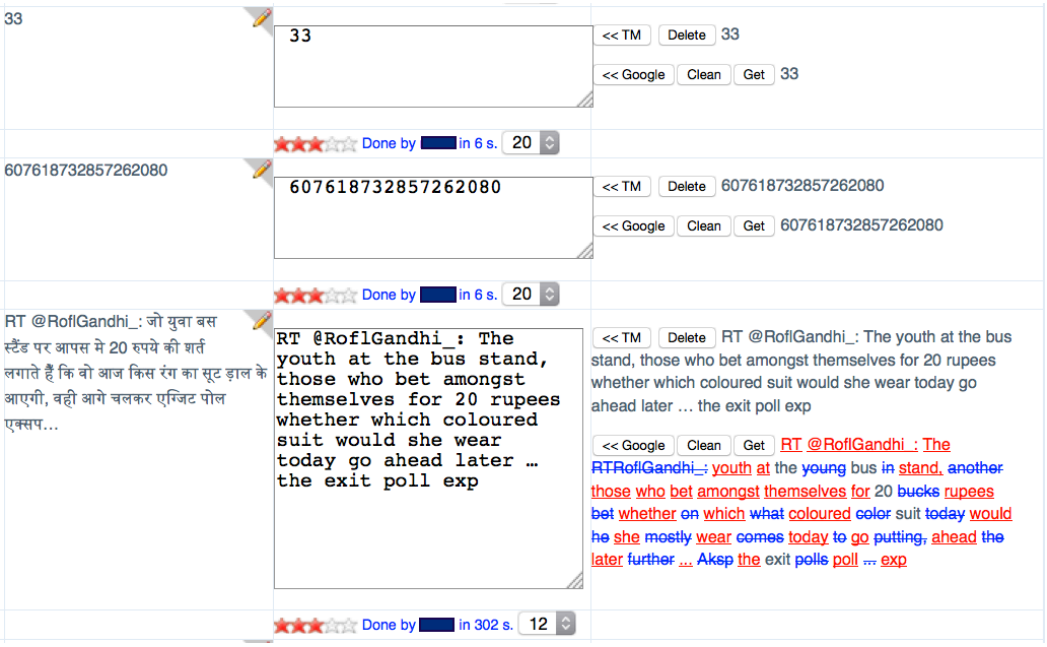

Figure 2: Screenshot of SECtra post-editing mode: source text, post-edit area with reliability level and quality score, post-editor's name (hidden in the image) and post-editing time. Right: trace of the edit distance algorithm computation.

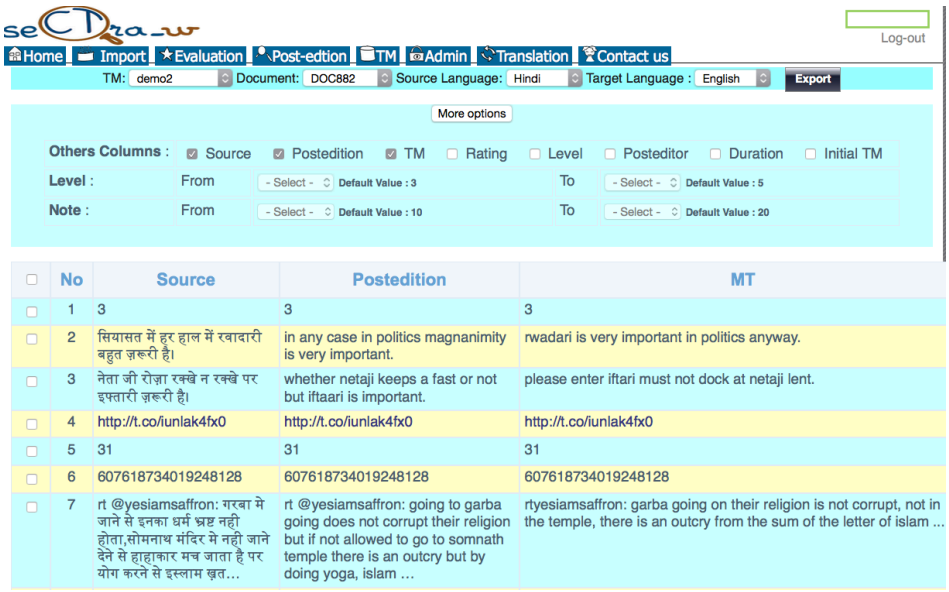

Figure 3: Screenshot of the SECtra Export mode. It allows data export in several formats. One also can view the translation memory. This figure shows only the source tweets, the pretranslations and the post-edition.

5 *Towards building a hi-en MT system useful for tweeters*

From the observations at hand, and knowing that specializing a MT system to a restricted sublanguage can dramatically increase all quality indicators (Chandioux, 1989) (Isabelle, 1987), we can outline a scenario to produce a specialized MT system that would be able to translate (fully automatically) 70% to 80% of the Hindi tweets into understandable English.

The idea is to include the *MT cross-lingual access* facility in the recommender system almost from the start, but not to make it accessible immediately, in order not to discourage forever tweeters to use it. There will be a phase whose length will depend on the number of bilingual Hindi-English speakers *contributing* to the building of a specialized hi-en tweet-MT system.

As has already been done successfully for French-Chinese (Wang and Boitet, 2013), we will estimate the *best size* $Size_{tw}$ of an aligned hi-en learning corpus (a first guess might be $Size_{tw} = 10000$ or $Size_{tw} = 15000$ for the observed sublanguage of Hindi tweets).

Initially, we will populate it using parts of some genuine hi-en corpus, if any, and, if none is available, a part of the CFILT[6] en-hi corpus. Even if inverted translations are notoriously not translation examples, an inverted parallel corpus is better than nothing. That will be the basis for building version 0.1 of a Moses-based specialized system, say, `twMT-hi-en-0.1`.

The *contributors* team will then post-edit what it can, working some time every day. Incremental improvement will be performed a certain number of times[7]after each new batch of *good enough* post-editions will become available, giving `twMT-hi-en-0.1` … `twMT-hi-en-0.20` if there are 20 incremental improvement steps. Version 1.1 (`twMT-hi-en-1.1`) will then be produced by full recompilation, and the whole process will be iterated.

[6]Centre for Indian Language Technology, IITB, India

[7]Experiments on French-Chinese have shown that improvement levels out after 10-20 incremental improvement steps. It is then necessary to recompile the full system, and that is also an appropriate time to modify the learning set by including all *good enough* post-editions, say, *Npe* bisegments, and keeping only $Size_{tw} - Npe$ of the *unspecialized* parallel corpus.

The PE interface will systematically propose the results of the current `twMT-hi-en-x.y` version in the *PE area* of each segment, but results produced by GT and if possible other systems (Systran, Bing, Indian systems) will also be visible, with a button to reinitialize the PE area with each of them. No development is needed, as this is a standard feature of the SECTra interface since 2008. The quality measure used to determine when the specialized system will be *good enough* to open the *MT cross-lingual access* facility to tweeters. There will be a first period during which `twMT-hi-en-x.y` will remain inferior to GT, that is, will require more PE time.[8]

After a certain version ($a.b$), the PE time for results of `twMT-hi-en-x.y` with $x.y \geq a.b$ will be less than that for GT, but results will still not be *understandable enough*. How to know if and when this will happen?

The experiment described in this paper shows that PE of current MT results allows to almost get to the required understandability level of 70%-80%, with a PE time of 12mn/p. We hope that MT outputs needing only 5mn/p of PE to reach 90% understandability will be *understandable enough* (70%-80%) *without PE*.

The idea is that, if that correlation holds, which we will verify by testing it every time a new version (x.1) is issued, we will open the *MT cross-lingual access* facility to tweeters when this *minimal understandability threshold* will have been attained through this *supervised learning* process.

Another worry will then be to ensure *non-regressivity*. It is expected that some continuous human supervision will remain needed, and that no *dedicated contributors group* will be maintainable. Then, some *self-organizing* community of contributors (post-editors) should emerge, somewhat like what has happened for many open source software localization projects. Another encouraging perspective is the announcement of a new kind of web service such as SYNAPS (Viséo, 2015), aiming at organizing contributive activities.

[8]The PE time for GT outputs will be estimated without any supplementary human work because experiments show a very good correlation between our *mixed PE distance* $\Delta_m(mt, pe)$ and $Tpe_{total}(mt)$.

6 References

Bibliografía

Boitet, Christian, Hervé Blanchon, Mark Seligman, and Valérie Bellynck. 2009. Evolution of MT with the Web. In *"Proceedings of the International Conference 'Machine Translation 25 Years On' "*, number from 2008, pages 1–13, Cranfield, November.

Chandioux, John. 1989. 10 ans de METEO (MD). In A Abbou, editor, *"Proceedings of Traduction Assistée par Ordinateur: Perspectives Technologiques, Industrielles et Économiques Envisageables à l'Horizon 1990: l'Offre, la Demande, les Marchés et les Évolutions en Cours"*, pages 169–172, Paris. Daicadif.

emojiList-1. 2015. Full emoji list. `http://www.unicode.org/emoji/charts/full-emoji-list.html`.

emojiList-2. 2015. Other emoji list. `http://www.unicode.org/Public/emoji/1.0/emoji-data.txt`.

GNIP. 2015. Gnip. `https://gnip.com/sources/twitter/`.

Huynh, Cong-Phap, Christian Boitet, and Hervé Blanchon. 2008. SECTra_w.1: An online collaborative system for evaluating, post-editing and presenting MT translation corpora. In *"Proceedings of the Sixth International Conference on Language Resources and Evaluation"*, pages 2571–2576.

Isabelle, Pierre. 1987. Machine Translation at the TAUM group. In *"Proceedings of Machine Translation Today: The State of the Art"*, pages 247–277, Edinburgh. Edinburgh University Press.

Viséo. 2015. Synaps website. `http://www.viseo.com/fr/offre/synaps`.

Wang, Lingxiao and Christian Boitet. 2013. Online production of HQ parallel corpora and permanent task-based evaluation of multiple MT systems: both can be obtained through iMAGs with no added cost. In *"Proceedings of the 2nd Workshop on Post-Editing Technologies and Practice at MT Summit 2013"*, pages 103–110, Nice, September.