A Machine Assisted Human Translation System for Technical Documents

Vishwajeet Kumar
Indian Institute of Technology
Bombay
Mumbai
vishwajeet@cse.iitb.ac.in

Ashish Kulkarni
Indian Institute of Technology
Bombay
Mumbai
kulashish@gmail.com

Pankaj Singh
Indian Institute of Technology
Bombay
Mumbai
pr.pankajsingh@gmail.com

Ganesh Ramakrishnan
Indian Institute of Technology
Bombay
Mumbai
ganesh@cse.iitb.ac.in

Ganesh Arnaal
SM Capital Advisors
Mumbai
ganesh.arnaal@smcapitaladvisors.in

ABSTRACT

Translation systems are known to benefit from the availability of a bilingual lexicon for a domain of interest. A system, aiming to build such a lexicon from source language corpus, often requires human assistance and is confronted by conflicting requirements of minimizing human translation effort while improving the translation quality. We present an approach that exploits redundancy in the source corpus and extracts recurring *patterns* which are : *frequent, syntactically well-formed, and provide maximum corpus coverage.* The patterns generalize over phrases and word types and our approach finds a succinct set of good patterns with high coverage. Our interactive system leverages these patterns in multiple iterations of translation and post-editing, thereby progressively generating a high quality bilingual lexicon.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Machine Translation**]: Natural Language Processing

General Terms

Machine Translation

Keywords

Mining, Machine Translation, natural language processing

1. INTRODUCTION AND RELATED WORK

The problem of language translation has been in focus for many decades and has seen contributions from both linguistic and computer science communities. Linguistic contribution [4] has come in the form of several language resources comprising of dictionaries, grammar and studies on units of translation. The statistical approach to MT itself falls under the general category of example-based MT (EBMT) [11] or memory-based MT [10]. These corpus-based approaches suffer from two major drawbacks - (1) parallel corpus is often expensive to generate and is often scarce or unavailable for certain language pairs or domains; (2) their quality of translation is not as good as that of human translation and therefore not suitable for certain applications like those involving translation of government documents or academic books.

Rule-based machine translation systems (RBMT) like Apertium [6] alleviate the need for a sentence aligned parallel corpus but require explicit linguistic data in the form of morphological and bilingual dictionaries, grammars and structural transfer rules. An in-domain (especially technical, legal) corpus often adheres to a certain lexical and syntactic structure and might be a good candidate for translation using rule-based systems. Grammatical Framework (GF) [8, 2] provides the necessary formalism to theorize rule-based translations and also provides a system to author abstract and concrete language syntax.

There is a body of work [5, 1, 3, 9] that studies the complementary contributions of humans and MT models and present "machine-centric" translation systems that leverage human input. These systems, referred to as computer aided translation (CAT) systems, typically employ a statistical MT model to translate text and provide a post-editing tooling to enable humans to correct the resulting translations. As an alternative to pure post-editing systems, interactive machine translation (IMT) [12] combines a MT engine with human, in an interactive setup, where, the MT engine continuously exploits human feedback and attempts to improve future translations. What constitutes the right unit of translation and how can the human feedback be incorporated in the underlying translation model, pose interesting research challenges.

We are further motivated by Frege's principle of compositionality [7], which states that the meaning of a compound expression is a function of the meaning of its parts and of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

K-CAP 2015 October 07-10, 2015, Palisades, NY, USA
Copyright 2015 ACM ISBN 978-1-4503-3849-3/15/10Ě ...$15.00.
DOI: 10.1145/2815833.2816961

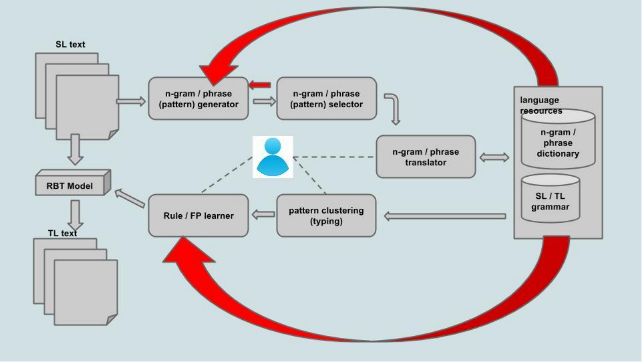

SL text
n-gram / phrase (pattern) generator
n-gram / phrase (pattern) selector
language resources
n-gram / phrase dictionary
n-gram / phrase translator
SL / TL grammar
RBT Model
Rule / FP learner
pattern clustering (typing)
TL text

Figure 1: System Architecture

the syntactic rule by which they are combined. Supplement[1] shows an example, taken from legal domain, of a compound expression and its constituent expressions. Some of these expressions comprise of categories that generalize over several tokens, thus, forming higher order recurring patterns in the corpus. Extraction of these patterns and using them as the unit of translation might enable us to better capture the structure and semantics of the domain.

We present an approach and a system that builds on these ideas to extract meaningful patterns from a domain corpus, gather human feedback on their translation and learn a rule-based translation system using the GF formalism. The system is "human-centric", in that, it heavily relies on manually curated linguistic resources, while the machine continuously prompts the human on *what* to translate. This interactive human-machine dialog produces a translation system that aims to achieve high precision in-domain translations and might find application in several technical domains including medical, education, legal etc. The system is available for demo at `http://mtdemo.hostzi.com/`.

2. SYSTEM ARCHITECTURE

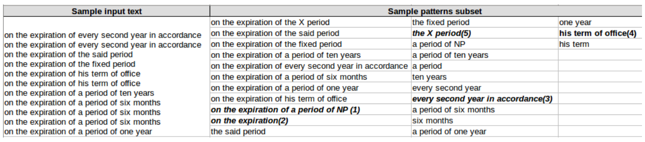
Our system (Refer to Figure 1) follows an iterative pipeline architecture where every component is modular. The system is interactive and takes human feedback on translations. The feedback is used to build linguistic resources and is incorporated into the underlying translation model. The translation model itself is expressed using the grammatical framework formalism, which is based on functional programming and type theory. This expressiveness and abstraction makes the model easily programmable by humans.

2.1 Pattern Extraction

This module aims to capture potential translation units present in the corpus. It takes as input an in-domain source language corpus and monolingual typed dictionaries and produces frequently occurring patterns as output. FPM Algorithm in appendix A explains our frequent pattern mining algorithm. Patterns might comprise of variable length *gaps*, represented by "X" or "_X_", which act as placeholders for generalized entities.

2.2 Pattern Selection

Pattern Extraction (Section 2.2) mines a large number of redundant patterns as potential translation units. Since getting manual translations for these candidate translation units is a costly operation, we identify a subset of patterns that are both diverse and maximally cover the in-domain source language corpus. The pattern selection algorithm in Appendix A provides details on this selection of a subset of "good" patterns, where, goodness of the subset is measured in terms of corpus coverage. Figure 2 provides an example, where, the first column contains sample text from a corpus and the other columns show the extracted patterns and the patterns (in bold) after the selection step.



Sample input text	Sample patterns subset		
	on the expiration of the X period	the fixed period	one year
on the expiration of every second year in accordance	on the expiration of the said period	***the X period(5)***	**his term of office(4)**
on the expiration of every second year in accordance	on the expiration of the fixed period	a period of NP	his term
on the expiration of the said period	on the expiration of a period of ten years	a period of ten years	
on the expiration of the fixed period	on the expiration of every second year in accordance	a period	
on the expiration of his term of office	on the expiration of a period of six months	ten years	
on the expiration of his term of office	on the expiration of a period of one year	every second year	
on the expiration of a period of ten years	on the expiration of his term of office	***every second year in accordance(3)***	
on the expiration of a period of six months	***on the expiration of a period of NP (1)***	a period of six months	
on the expiration of a period of six months	***on the expiration(2)***	six months	
on the expiration of a period of six months	the said period	a period of one year	
on the expiration of a period of one year			

Figure 2: Sample patterns after pattern selection

2.3 Pattern Translator

The translator module involves humans providing translations of patterns selected by the pattern selection module. These are used to build a bilingual lexicon and along with monolingual dictionaries are used to train a rule-based translation engine. In subsequent iterations, the trained translator provides multiple translation options that the user can choose from or override. Section 2.6 offers further details on the user interface.

2.4 Generalization of Translation Units

This module helps in generalizing the translation units by clustering them together. This further aids in reducing the number of rules in subsequent grammar generation, where, each cluster identifier acts as the non-terminal of all productions involving the cluster members.

We have seen in various sentences that the phrases whose internal reordering is same also have the same external reordering when translated to other language. So we clustered translation units having same internal reordering into one cluster. We used reordering distortion score between translations of two translation units as a measure to cluster translation units . Since cluster represents all the translation units present in that group, it also represents their translation behavior. That is, their external/internal reordering is same. Same external reordering helps makes it possible to write a single translation rule for all the member translation units of a cluster.

2.5 Rule / FP Learner

Once translation units are extracted, selected, translated and stored in language resources database next job in hand to annotate sentences with translation units or in other words represent sentences in the form of sequence of translation units. If the coverage for a sentence is 100% whole sentence can be represented only with the help of translation units. Once sentence is represented in canonical form we parse and linearize it using grammatical framework rules.

Grammatical framework (GF) is an extension of logical framework with a component called *concrete syntax*. Reordering rules and rules for handling gender, number and

[1] `http://www.cse.iitb.ac.in/~vishwajeet/kcap_example.html`

person information while doing look-ups is written using the GF formalism. The main purpose behind using grammatical framework is its functional nature, also grammatical framework has a concept called abstract syntax which provides interlingua representation. Interlingua representation helps in linearizing in different languages very easily just by writing concrete grammar for that language.

2.6 System User Interface

Our system has an interactive user interface for humans to translate patterns and n-grams. It also has provision for expert users to configure pattern length and frequency threshold for pattern extraction. Different user interface features are shown at `http://http://www.cse.iitb.ac.in/~vishwajeet/kcap.html/`. Figure 5 on the link depicts the features provided to expert users. Users can upload a new corpus using the *Upload Input File* option marked with label 1 in the figure. The *Upload Dictionary* option (labeled 2) enables users to upload bilingual dictionaries for the system to perform lookups and provide translation suggestions. Users can either choose to run the system and extract patterns on the optimized default configuration (labeled 3) or they can manually configure the pattern length and frequency (labeled 4).

Once patterns are extracted, filtered and validated by the system, users use the web-based system shown in Figure 1 for providing translation feedback. Human translators are shown the current sentence (Refer to label 1 in the Figure 1) along with the previous and next sentences as context information. Patterns are displayed below column labeled as *fragment* (Refer to label 2 in the Figure 1). On hover over patterns or untranslated n-grams, the span covering that pattern or n-gram in the sentence gets highlighted (refer to Figure 4). For patterns containing generalized non terminals (labeled 2), translators can view all the instances (label 4 in the figure) of non terminals (NT) by hovering over the NTs. Initially a translation of patterns and untranslated n-grams (labeled 5 in the Figure 1) is suggested by the system using translated patterns database, glossary lookup and SMT. Translators can even configure the source for getting the suggestion (a) they can choose to get translation suggestion from SMT system by clicking on SMT button (label 12 in Figure 1) or (b) they can choose to get translation suggestions from database by clicking on glossary button (label 11 in Figure 1). Translators can edit the translation suggestion (labeled 3 in Figure 1) given by the system and correct them. They can also reorder the composed translation of sentence by clicking on reorder button (labeled 6 in Figure 1), which presents a simple drag and drop interface to the user (refer to Figure 6). Finally, if users wish to edit the composed translation they can do that by clicking on final editing button depicted by label 7 in Figure 1. After final editing users can save the translation by clicking on the save button (labeled 8). Users can also download the translations by clicking on download button (labeled 9). In order to get translation suggestion for a particular word or phrase, user can enter the text in suggestions panel on the right and get multiple translation suggestion for the particular word or phrase.
Important Features of the system:-

- Once a translator translates a pattern, a pattern instance or an n-gram, the system auto-translates it the next time it appears in a sentence.
- If a pattern, pattern instance or n-gram is translated differently in different sentences, the system lists all of them as choices for the user to decide or enter a new translation.
- The system also has an integrated suggestion component that fetches translation suggestions from various sources. Users can use this to get translation suggestions for words or phrases and choose the best translation from the choices.

3. EVALUATION

We evaluate the system in terms of the quality of extracted patterns, GF grammar and system efficiency. Evaluation was done on five public datasets *viz.* the Constitution of India[2], Spoken Tutorial[3], NCERT Biology[4], Income-tax Act[5], and NCERT Physics[6]. These datasets belong to the domains of government documents, technical tutorials and academic books, where, high quality translations are an imperative. Table 1 shows the corpus statistics in terms of number of sentences for each of the datasets.

3.1 Number of Frequent Patterns and Corpus Coverage

The number of frequent patterns increases with the size of the corpus. Corpus coverage depends on the number of patterns extracted from a corpus which adhere to specified pattern length and frequency. Table depicts information about number of filtered patterns extracted and coverage on five different corpora.

Table 1: Datasets and corpus coverage by patterns

Domain	#Sentences	#Frequent Patterns	#Frequent Instances	Coverage %
Constitution of India	1582	12946	154218	86.62
Spoken Tutorial	16233	32974	10846	78.32
NCERT Biology	1144	615	12407	60.82
Income-Tax Act	1758	8391	104998	89.34
NCERT Physics	8013	15070	244034	79.94

3.2 Effect of Varying Pattern Length and Frequency Threshold for Pattern Extraction

One of the criterion to assess the quality of an individual extracted pattern is whether or not it appears in unseen data, thereby covering sentences in that data. A set of such patterns is then considered to be “good” if it collectively offers a high coverage on an unseen data. We split the datasets into MINE and TEST, where, the MINE split was used for extracting patterns and their coverage (in terms of number of words covered) was evaluated on the TEST split. We perform three-fold cross validation, varying both pattern length and frequency threshold from 2 to 6 and report coverage on

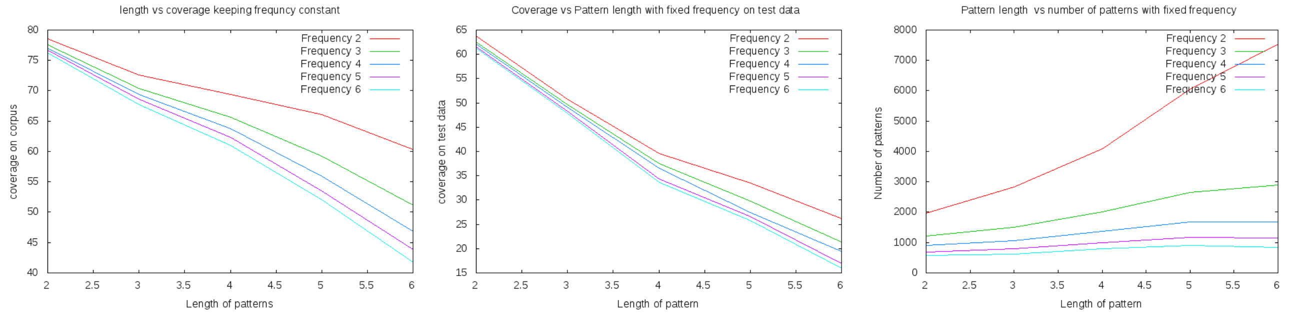
[2] `http://indiacode.nic.in/coiweb/welcome.html`

[3] `http://spoken-tutorial.org/`

[4] `http://www.ncert.nic.in/NCERTS/textbook/textbook.htm?kebo1=0-22`

[5] `http://www.incometaxindia.gov.in/pages/acts/income-tax-act.aspx`

[6] `http://www.ncert.nic.in/NCERTS/textbook/textbook.htm?leph1=0-8`



(a) Coverage vs. pattern length on the mining data

(b) Coverage vs. pattern length on the test data

(c) Pattern length vs number of patterns for a fixed threshold

Figure 3: Corpus coverage for varying pattern lengths and frequency thresholds

MINE and TEST sets. Figure 3 captures the trade-off between pattern length, frequency threshold and coverage. For a fixed threshold, as the pattern length increases, the coverage on both MINE and TEST sets progressively decreases. Same observation applies when we fix the pattern length and increase the frequency threshold. We also observe that the gap in coverage is much smaller for varying frequency thresholds at smaller lengths and this gap progressively widens as the pattern length increases.

3.3 Effect of Varying Dictionary Size on Corpus Coverage

Our pattern selection algorithm constrains the cardinality of the set of patterns while maximizing the corpus coverage. This corresponds to limiting the size of the bilingual dictionary and this is desirable as the size of the bilingual dictionary is proportional to the human effort for translation. The corpus coverage increases with increasing size of the dictionary, however this increase is not linear but rather flattens with increasing size of the dictionary. Figure 4 captures this relationship between coverage and fraction of patterns selected after sub-setting for different datasets.

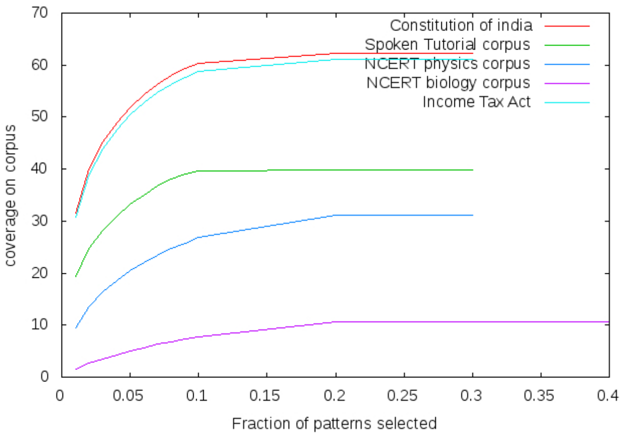


Figure 4: Coverage vs number of pattern selected after pattern selection

3.4 Induced GF grammars

Once users provide translations of patterns, their instances, uncovered n-grams in sentences and reorders different chunks, grammatical framework rules are induced. Firstly, abstract syntax is induced which defines what meanings can be expressed in the grammar and then concrete English and concrete Hindi syntax is induced which provides mapping from meanings to strings in English and Hindi languages. Figures 8, 9 and 10 on the link[7] illustrates sample induced GF grammar. For a new sentence, extracted and translated patterns are given as input to GF grammars and if a match is found, then the sentence is reordered using the mapping from the concrete syntax.

4. CONCLUSION

We presented an interactive machine translation approach for high quality translation of technical domain corpora. Given an in-domain source corpus, our system mines minimal number of frequent patterns that maximally cover the corpus. Leveraging humans for their high quality translations, we continuously rebuild a rule-based translation engine that is realized using GF formalism.

5. REFERENCES

[1] V. Alabau, C. Buck, M. Carl, F. Casacuberta, M. Garcıa-Martınez, U. Germann, J. González-Rubio, R. Hill, P. Koehn, L. Leiva, et al. Casmacat: A computer-assisted translation workbench. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28, 2014.

[2] A. S. A. R. L. M. M. G. Cristina Espa˜na-Bonet, Ramona Enache. Patent translation within the molto project. *MT Summit*, August 2011.

[3] M. Denkowski and A. Lavie. Transcenter: Web-based translation research suite. In *AMTA 2012 Workshop on Post-Editing Technology and Practice Demo Session*, page 2012, 2012.

[4] B. J. Dorr. *Machine translation: a view from the Lexicon*. MIT press, 1993.

[5] M. Federico, N. Bertoldi, M. Cettolo, M. Negri, M. Turchi, M. Trombetti, A. Cattelan, A. Farina, D. Lupinetti, A. Martines, et al. The matecat tool. In *Proceedings of COLING*, pages 129–132, 2014.

[7] `http://http://www.cse.iitb.ac.in/~vishwajeet/kcap.html/`

[6] M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144, 2011.

[7] F. J. Pelletier. The principle of semantic compositionality. *Topoi*, 13(1):11–24, 1994.

[8] A. Ranta. *Grammatical framework: Programming with multilingual grammars*. CSLI Publications, Center for the Study of Language and Information, 2011.

[9] J. Roturier, L. Mitchell, and D. Silva. The accept post-editing environment: A flexible and customisable online tool to perform and analyse machine translation post-editing. In *Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice*, pages 119–128, 2013.

[10] S. Sato and M. Nagao. Toward memory-based translation. In *Proceedings of the 13th conference on Computational linguistics-Volume 3*, pages 247–252. Association for Computational Linguistics, 1990.

[11] H. Somers. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157, 1999.

[12] A. H. Toselli, E. Vidal, and F. Casacuberta. Interactive machine translation. In *Multimodal Interactive Pattern Recognition and Applications*, pages 135–152. Springer, 2011.

APPENDIX

A. ALGORITHMS

Algorithm 1: FPM algorithm

```
Data: Corpus $C$, Pattern length $L$, Frequency threshold $T$,
      Maximum consecutive gaps of tokens $G$
Result: Set $F$ of frequent patterns
Maintain a dictionary structure globalPatternList where key is
pattern and value is list of span
for each sentence $s$ in $C$ do
    maintain an array of list, slist, of size $|s|$, such that, $slist[i]$
    stores all one length pattern along with its span in the
    sentence which starts from $s_i$
    using slist, create a 2D array of list, smatrix, of size $|s|xL$
    such that, $smatrix[i][j]$ stores all patterns, along with its
    span in $s$, which starts from $s_i$ and of pattern length $j$
    Filter pattern from smatrix whose span is syntactically
    incomplete
    Add these patterns to globalPatternList
end
Initialize patternWithGap dictionary
for $i$ in $1 \cdots L$ do
    for valid mask $v$ of length $L$ do
        for pattern $p$ of length $i$ in globalPatternList do
            apply v on pattern p and create a new pattern $\hat{p}$
            if $\hat{p}$ is present in patternWithGap then
                update its spanlist by doing union with span
                list of p
            else
                add $\hat{p}$ in patternWithGap with its spanlist as
                spanlist of p
            end
        end
        remove patterns of length $i$ and with gap position
        according to mask and having spans count less than $T$
    end
end
remove patterns from globalPatternList whose number of spans
is below T
output $patternWithGap \cup globalPatternList$
```

Algorithm 2: Pattern Selection

```
Data: Dictionary of patterns $P$ with its spanslist, Number of
      words in corpus $N$, Max size of selected set $k$
Result: Set $F$ of diverse and high coverage (in terms of words)
        patterns
$F = \emptyset$ , $bitCorpus \leftarrow \emptyset$
for $i \leftarrow 1$ to $N$ do
    $bitCorpus[i] \leftarrow false$
end
for $i \leftarrow 1$ to $k$ do
    $currentBest \leftarrow NULL$ , $currentBestCover \leftarrow 0$
    for each pattern $p$ in $P \setminus F$ do
        $cover_p \leftarrow$ count of false bits in bitCorpus which is in
        spanlist of p
        if $cover_p > currentBestCover$ then
            $currentBest \leftarrow p$ , $currentBestCover \leftarrow cover_p$
        end
    end
    if $currentBest$ then
        $F \leftarrow F \cup$currentBest
        $BitCorpus[i] \leftarrow true$, if i is in the spanlist of
        currentBest
    else
        break
    end
end
output $F$
```