Automatic Detection of Acronyms in Hindi texts

Anubhav Bimbisariye 11131
Kanishk Varshney 11350
Instrustor Incharge: Dr. Amitabha Mukerjee
{anubhav, varskann, amit}@cse.iitk.ac.in
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

ABSTRACT

Acronym detection for common Hindi text encountered daily has not yet been tried to the best of our knowledge. Amongst all the types of acronyms that can be found in a Hindi text, we present the most abundant types with an analysis on other types as well. Our analysis shows that majority of these acronyms have a definite pattern, or expressions, and so, can be detected using an identification rules approach. Another type of acronyms are detected using a common word elimination approach with the help of a dictionary. Our methods have yielded a precision and recall of 89.1% and 90.9% respectively.

INTRODUCTION

General Hindi texts encountered in daily life in today's date are newspapers, Wikipedia articles, online Hindi pages, books etc. In majority of industry and office work, English has been standardised in India, and so Hindi is encountered mostly through these common means, and not in business documents, or official works, except some government offices.

We encounter a lot of acronyms in these daily texts such as shortened names of educational institutes like आईआईटी, एनआईटी, डीपीएस, or Political Parties like भाजपा, बसपा, etc.

Then there are acronyms like कि.मी. , standing for a "*Kilometre*". These are some of the most abundant and a little easy to understand acronyms. However, there can be presence of some ambiguous acronyms like आप, which can stand for *"Aam Aadmi Party"* or *"you"*.

Acronyms are a recent addition to languages. They make our work and communication faster, and easier. However, this is the case only when the reader is familiar with them.

Otherwise, they are bound to slow down the understanding of text as the reader or user tries to decipher it.

Even though they are a recent addition to the linguistics, there rising number and abundance makes it an important problem to automatically detect the present acronyms. It might prove to be a lot of help to various problems like OCR and recognition, semantic analysis etc.

Our analysis of algorithm takes inspiration from, and considers a lot of other methods which have previously been used for different languages, and majorly, in English.
We analyse the types of acronyms found in different languages, and the approach taken by others to solve them, to decide our own route for Hindi.

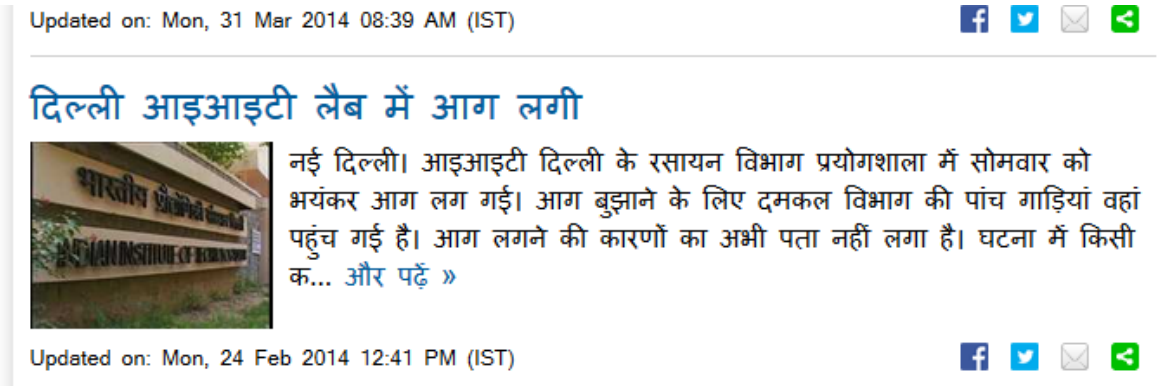

Image is a snip of http://www.jagran.com/

RELATED/PAST WORK

In the past dozen or so years, a lot of work has been done towards acronym detection by various people. Various methods used are based on heuristics, Machine Learning (ML), rule based definitions and statistics of document or corpora. Many approaches use a “Stop word” list in order to handle all the troublesome cases.

Yeates[2] introduced a Three Letter Acronym system in a digital library context. Heuristics approach is implemented to match an uppercase SF with a closely located long form. His methods yield him a recall of 93% and a precision of 88%.
Park and Byrd use identification rules for acronyms, linguistics hints and text markers. They also integrate the detection of acronyms containing digits.

Background

Definition: An acronym is an abbreviation formed from the first letter of a group of words, with group size ranging from two to 5-6 words, and in rare cases, even more. Though this is the standard definition of an acronym, however the style of acronyms has been modified a lot and in present times, there are a variety of acronyms that can be encountered like ARPANET in which NET stands for 'Network', so instead of one letter from this word, 3 have been taken. P2P means 'peer to peer' where 'to' has been replaced with a homophone two, i.e. '2'. 'i.e.' is read as 'That is', however is derived from a different word 'id est'.

Features of Acronyms

Acronyms are made in different ways. First, if we look at English, some examples mentioned above are of acronyms like P2P, id est- i.e. and ARPANET. There are some acronyms like 'radar', which are standard acronyms, but are not exactly a short form of type first letters from words.

Such cases of acronyms are very troublesome, and they ask for a ML based algorithm or a Heuristics based approach. Some almost always require a long form to be present either implicitly or explicitly in the document, so that the short form may be identified as a valid acronym.

French can present even more complex acronym and so, Menard and Ratte[1] present a classifier based approach for acronym detection.

When we look at Hindi acronyms, we find that in Hindi, acronyms are even a more recent addition than in English and some other languages. Hindi has few troublesome cases. Our analysis shows us the following types of acronyms present in Hindi:

Type	Example	Information	Estimated Difficulty
English Based	IIT - आईआईटी	English abbreviation translated letter to letter as written in Hindi.	Moderate
Short Form	भाजपा- भारतीय जनता पार्टी	Hindi Abbreviation.	Difficult
Spoken Short form	आप- आम आदमी पार्टी	Merged syllables from आआप based on sound.	Very difficult
Full Stop Words.	कि.मी.	Most often, abbreviation of actually English words, like kilometre is an English word.	Easy.

Words, which are acronyms in English, like 'radar', written as it is in Hindi are considered to be words, and not acronyms.

Acronyms may be present in a text as an explicit declaration like- IIT(Indian institute of Technology) or Indian Institute of Technology (IIT).
It may be present as a semi explicit form. Like Indian institute of Technology, commonly known as IIT.
Or, it may be present as implicit form i.e. IIT mentioned somewhere, Indian Institute of technology mentioned elsewhere without any clear connection.
Finally, it may be present without any long form at all.

Out of these 4 types, we found that the first 3 types of declarations were very rare, the first and third type being 1 in about 200 acronyms, and 2nd type, rare enough to be not present in the corpus.

Approach

Based on our analysis of the corpus, we realised, that most of the acronyms were of the 4th type, that is, present without a Long Form (LF). This posed a difficulty for methods based on validation of candidate Short Forms (SFs) by searching for their LFs. Such searches require tedious methods like heuristics, pattern matching, allowing of certain kinds of errors, considering syllable merging, as in the case of आप. These kinds of acronym mean that we cannot directly just try to split a candidate SF and look for its possible LF. In this case, such a method would have yielded a pair आप- आदमी पार्टी instead of आम आदमी पार्टी which is not desirable if effort is being made to detect the correct acronyms. So, apart from these rare cases, it is hard to definitely match the SFs for a LF, given that there are about 1 in 200 SF-LF pairs in the corpus. This motivated us to recognise SFs without the need to look at the possible LFs. We look for ways to do this. We find that the 4th type of acronyms are made majorly from English based acronyms, and full-stop words. We can detect such words easily using Regular expressions.

Regular expressions

All words made of English letters translated letter to letter in Hindi. Letter correspondence:

A- ए	B- बी	C- सी	D- डी
E- ई	F- एफ	G- जी	H- एच
I- आइ, आई	J- जे	K- के	L- एल
M- एम	N- एन	O- ओ	P- पी
Q- क्यू	R- आर	S- एस	T- टी
U- यू	V- वी	W- डब्ल्यू, डब्लू	X- एक्स
Y- वाय	Z- ज़ेड		

Any English based acronym in Hindi is composed of a combination of these words. So, a natural regex which accepts all such words is:

R= (A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q+R+S+T+U+V+W+X+Y+Z)*.

We attempt to search for this regex in it. If a word or text consist of anything which this regex accepts, we consider it an acronym.
However, it was found that the first try with this regex did not yield good result, when tried specifically with a corpus composed of exclusively these type of acronyms. The reason being, this regex takes up words like मुघल-ए-आज़म, because of presence of 'ए' in it. It took all words with any single instance of these letters as an acronym.
A noticeable point from the above work was that none of the single letter word, or those which matched due to single letter like 'ए'क्कुपंचर्र were acronyms. This is because Hindi doesn't have single letter acronyms. So, the regex was changed to

R= (A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q+R+S+T+U+V+W+X+Y+Z)+

This resulted in an improvement, and all single letter false positives were eliminated. However, we found that some acronyms were missing in both the approaches which do come in the type of acronyms we were solving for.
These are two types, in one, the standard correspondence for English letter was not followed, but a different form. Eg: E- आई, as in IIT in some place places changed to E- आइ. Similar cases with W etc. were found. To solve this, simply multiple forms of a letter were introduced in the regex.

Another type was like आई.आई.टी. , which weren't matched because they don't match the regex once it is constrained to look for two or more lettered acronyms. This problem was solved by considering that these acronyms will be discovered in the detection of acronyms made with full-stops, like कि.मी. as all these words also consist of more than one letter or full-stop.

Stop words: These were detected using regular expression

R= (\.)*(\.)

This regex matches any word which has full stops in it.
Words which are not acronyms but have full stop in them are extremely rare in Hindi. This means, that the probability that the word matched with this regex is an acronym is ~1.

For the SF words of Hindi, we considered 2 approaches, out of which we could implement only 1 within the time scope of the project.

Method 1: breakup of each Candidate SF in Hindi letters, and search for a matching LF. This task is tedious, and needs a lot of considerations as mentioned above due to various differences in the ways acronyms are defined and modified.

Method 2: Dictionary Word elimination. This is the method we have applied. We take all the words which belong to dictionary and eliminate them from the corpus. This leaves us with words which are not recognised by any dictionary in Hindi. We use Wordnet[3] database for this purpose. The words which are left are have a very high possibility of being an acronym. This technique brings out the acronyms like names of political parties. Basically, the words which are Hindi short forms are detected. However, this approach, though detects a lot of acronyms which were missed, it also brings out many more words which are actually not acronyms. These include names like Puneet, Rahul, and words like plane, current, websites, etc. This has drastic effects on performance, degrading it to a precision of about 60%. This is undesirable. In order to correct this, a possible option is to implement a lighter version of Method 1, and take its candidate SFs from this method. Another possible option is to modify the dictionary, and build it our self by inserting high frequency words from a giant

corpus and using a semantic analysis on the text to decide which words can potentially be an acronym. POS tagging in Hindi corpus, will surely help.
Combining these acronyms, we get our final result, however, there is one problem.

Problem: Common Hindi text is not independent, and often contains a garbage of texts and symbols from maths, English, urls, normal style writing with double dots etc. this causes words like 'us..' to be extracted from 'so this was done by us..' due to 2 dots in it, which match the regex. This is eliminated by tweaking regex to remove words with continuous dots.
English words: Words like "D.P.S", "www.somewebsite.com", 'where' etc.
Numbers: 1.23, 29, etc.
These are eliminated by checking detected word to contain alphanumeric characters.
A separate script is run to filter the complete results for this garbage.
Another problem is that often in corpus, the words are not properly placed, they do not always have spaces in between them, they are continuous, they start right after 'poorna viram' etc.
A script to modify the corpus for all words is run for this. It organises the corpus properly, and eliminates ambiguities.
Some technical problems encountered were problems with using regular expressions for Hindi. It was found, regular expressions don't work for Hindi exactly as they do for English.

Earlier approach: Before trying out the identification rules based methods, we tried to use English acronym detection methods and find out acronyms from a parallel English-Hindi corpus. However it wasn't a success due to certain issues.
1- The code for English acronym detector is not available as it has been commercialised or patented and is no longer open source.
2- Even with acronym detector, if it had been available, we would need to do a lot of semantic analysis and word matching to relate English words with the Hindi words and find out which Hindi word matches with the English acronym. It might have been relatable in terms of basic structure, or letter wise reliability, still it would have needed a lot of analysis.
3- The parallel corpus available were not very good. They hardly had any acronyms present in it, as we manually figured out by having a look at corpus.
So, work for it seemed futile.

Future Work

Due to Constraints of the course, and semester management, we could analyse and research on some aspects of Hindi based acronyms. However we wish to test a few other approaches, algorithms, and methods that we are hopeful about. We are of opinion that a simplified version of a few of these algorithms will help a lot. These include pattern matching technique for possible SFs to look for a LF within some error limit and range. Another approach can be to use ML approaches and build a stopword list for troublesome and common acronyms manually. We would also like to try yeates' method based on n-gram analysis.

Results

Corpus used: due to uselessness of the parallel corpus, we built our own corpus by taking text from:

1. www.jagran.com/
2. www.bhaskar.com/
3. navbharattimes.indiatimes.com/
4. https://hi.wikipedia.org/

Test corpus word size: Approximately 10000 words.
Corpus was *hand tagged*, and consisted of 193 actual acronyms.
Correct acronyms detected: 175
Total acronyms detected: 195
Recall: 90.9 %
Precision: 89.74 %
These results also consider the acronyms that we didn't aim to detect.

Conclusion

Acronym detection method we have implemented has yielded results almost similar to the paper we have been following as a guide, i.e. Menard and Ratte's classifier based approach. However we are optimistic about the scope by which we can further improve our results. We hope to implement advanced methods mentioned in future work with more pleasing results, and present this approach as a base to start research in this field for Hindi.

Acknowledgements

We would like to express our sincerest, heartfelt gratitude to our advisor and mentor for the project, also our instructor incharge, Dr. Amitabha Mukerjee, for showing us the path to bring this project to a conclusion, for mentoring us, and to motivate us at times we faced difficulties. This project has showed us the fun in the field of Artificial Intelligence and Natural Language Programming. The course has overall been very fun and enjoyable, and we hope we are able to contribute a little to the field through our project.

References and Guide.

[1] Pierre André Ménard & Sylvie Ratté (2010) “Classifier-based acronym extraction for business documents” © Springer-Verlag London Limited 2010.

[2] Yeates, S. (1999), Automatic extraction of acronyms from text. In Third New Zealand Computer Science Research Students' Conference, pages 117-124

[3] Dipak Narayan, Debasri Chakrabarty, Prabhakar Pande, and Pushpak Bhattacharyya. An experience in building the indo-wordnet - a wordnet for hindi. In GWC'02: Proceedings of the First International Conference on Global WordNet, Mysore,India, 2002