Stochastic Modelling and Computational Sciences

THE DEVELOPMENT OF A RULE BASED LEMMATIZING ALGORITHM FOR DHUNDHARI LANGUAGE

Varda Pareek[1] and Nisheeth Joshi[2]
[1]Department of Computer Science, Banasthali Vidyapith, Rajasthan-India
[2]Speech and Language Processing Lab, Centre for Artificial Intelligence, Banasthali Vidyapith, Rajasthan-India
varda2976@gmail.com[1] and jnisheeth@banasthali.in[2].

ABSTRACT
Stemming and lemmatization are two processes to analyze a language. Stemming is the process to extract root word but sometimes stemming suffers with problem of over stemming and under stemming where over stemming is resolved by lemmatization. In this paper a rule-based lemmatizer for Dhundhari language is developed and its process is discussed critically. Dhundhari is an extremely low resource and very rich morphological language. Dhundhari language is widely spoken in Jaipur region of Rajasthan state of India.In this paper more than 200 rules were created for lemmatization. The Inflectional and derivational lemmatizers were evaluated separately. The inflectional accuracy was 99.2% and derivational accuracy was 99%. Overall combined accuracy (Inflectional+ Derivational) was 99.1%.

Keywords: Stemming, Lemmatizer, Under-stemming, Over-stemming, lemma, Stem, Rule-based.

1. INTRODUCTION
Natural language processing (NLP) is a field of computer science that provides the ability to machine for understanding text and words in the same way a human being can. Deep learning and machine learning are popular technologies to provide linguistic knowledge to machines. These technologies help the machine to understand human language and its sentiments. Huge number of languages are spoken around the world and every language is different from another language in some way.

Dhundhari is one of the six commonly spoken Rajasthani languages. It is widely spoken by the people residing in Jaipur region of Rajasthan state of India. It comes under the extremely low-resource language category where hardly any dictionary or grammar is available. But there are many Dhundhari-speaking people all over the world. The word structure must be taken care for changing one language into another language. Examination of this comes under the study of morphology.

The study of the structure of words is known as morphology which analyses the word structures and parts of words like stems, root words, prefixes, and suffixes. To understand the word structure of a language, morphological analysis is needed. Stemming and lemmatization are common techniques to do morphological analysis. Stemming is a process to extract a root word by removing affixes but sometimes the stemming process suffers with under-stemming. To solve the problem of under-stemming, a process of lemmatization is done. Lemmatization is a process that converts the word into its meaningful base form and is known as a Lemma. In this paper, a rule-based lemmatization algorithm for Dhundhari language has been developed for the first time. For both Hindi and Dhundhari lemmatizing rules are developed as Dhundhari and Hindi carry many common words. This has been bit challenging in development of lemmatizer for Dhundhari

2. LITERATURE REVIEW
Plisson et al. (2004) developed a Ripple Down Rules (RDR) approach for word lemmatization problems from Slovene-free text and higher accuracy results were observed.

Perera and Witte (2005) proposed German nouns lemmatization algorithm where the construction of lexicons was automatic, and it was concluded that this algorithm produced accurate results.

Zhang and Sumita (2007) presented a linear interpolation method for Chinese – English language pair. They integrated the translation model's homologous features of the lemmatization and non-lemmatization system, and efficient performance was observed.

Al-Shammari and Lin (2008) presented the first Arabic Lemmatization method and compared its performance with the Khoja stemming method. It was observed that the lemmatization method outperformed the stemming method in word normalization for the Arabic language

Jongejan and Dalianis (2009) developed an affix algorithm for morphological changes by training lemmatization rules. It was concluded that the affix algorithm outperformed the suffix algorithm

Paul et al. (2013) did work on the Hindi lemmatizer where the approach was rule-based.This lemmatizer was created for optimization. It was seen that the system produced 89.08% accurate results.

Rodrigues et al. (2014), in their paper, presented a tool of cross-platform lemmatization for the Portuguese language which was developed in Java. It was observed that the system produced 98% accurate results.

Strakova et al. (2014) discussed POS- and NE taggers for the Czech language. It was observed that the results were nearly similar for the three Czech systems, Morce, Featurama ˇ, and MorphoDiTa.

Yeog et al. (2016) investigated the use of the English-Malay Dictionary and lemmatizer of English for improving the English-Malay translation's BLEU score. It was seen that by using the Stanford parser, improvement was higher.

Lo et al. (2016) proposed the NRC submission to the WMT 2016 Russian-English news translation task. They used Russian lemmas for improving word alignment and used neural network joint models and lexical translation models. The system was based on semi-supervised learning and efficient performance was observed.

Bermanis and Goldwater (2018) proposed Lematus based on the NMT of Sennrich for learning context-sensitive lemmatization and concluded that higher lemmatization accuracy was measured for low-productivity languages. It was seen that morphological regularity played a major role.

Manjavacas et al. (2019) proposed a joint learning approach that was based on a bidirectional Language Model for improving lemmatization. it was concluded that LM loss helped in capturing morphological information.

Ingólfsdóttir et al. (2019) focused on the Nefnir lemmatizer for Icelandic which used suffix substitution rules. It was seen in the results that Nefnir obtained higher accuracy for PoS-tagged input.

Kanerva et al. (2021) studied and proposed a sequence-to-sequence lemmatizer. it was seen after the evaluation of the lemmatizer on 52 different languages and 76 different treebanks, that this system was better than all the latest baseline systems.

Dhar et al. (2022) proposed PT-Inflect for overcoming parallel data scarcity for morphologically rich languages under the low resource and it was seen that this technique outperformed the NMT.

Shaukat et al. (2023) focused on the Urdu lemmatization system which was based on a dictionary-based approach and manually developed a dictionary where words contained POS tag and lemma. Data was collected from the mono corpus and Wikipedia dump and the PoS tag and the lemma relationship were investigated. By applying PoS Tagging Phase and Lemma Generation Phase cases, it was seen 66.79% and 76.44% accuracy respectively.

Salih et al. (2023) worked on the morphological production and analysis of Arabic nouns. They used e stem-based method and the NooJ toolkit. Efficient results were achieved in the experiment.

Hafeez et al. (2023) proposed a lemmatization method that was based on recurrent neural network models for the Urdu language. It was seen this approach handled proper nouns, inflectional and derivational morphemes, stop words, loan words, and discretized Urdu words.

3. Study of Dhundhari Language

Dhundhari (likewise alluded to as Jaipuri) is an Indo-Aryan language that is largely spoken in the northeastern part of 'Rajasthan" territory, India. This area is named the Dhundhar area. Jaipur, Sawai Madhopur, Dausa, Tonk and a couple of parts of Sikar constitute area where Dhundari-talking individuals can be seen. According to Indian Census 2011, there are 1,476,446 native speakers of Dhundhari in Rajasthan. It keeps on being spoken widely in and surrounding part of the capital of Rajasthan (Jaipur).

It has been already argued that the morphology which analyzes the word structures and parts of words like stems, root words, prefixes, and suffixes. In morphology, Morpheme is the smallest and the most meaningful unit. There are two types of morphemes- Inflectional and Derivational. Inflectional morphemes don't change the part of speech of a word whereas part of speech of a word changes in derivational morphemes change.

Dhundhari is a morphologically rich language. There are 31 Consonants, and 10 vowels present in Dhundhari.

Consonants in Dhundhari:

क ख ग घ च छ ज झ ट ठ ड ढ ण त थ द ध नव प फ ब भ म स य र ळ ल ह श

Vowels in Dhundhari:

अ आ इ ई उ ऊ ए ऐ ओ औ

The construction of Dhundhari is very like to the language Hindi. The subject-object-action sequence sentence is occupied in Dhundhari. Most interrogatives used in Dhundhari are not similar in Hindi. Nonetheless, components of punctuation and central phrasing shift to the point of genuinely hindering shared coherence. Few expressions of Sanskrit are utilized in Dhundhari. In Dhundhari words can also be separated into lexical classes like thing- 'noun', action word – 'verb', modifier– 'adjective' and intensifier- 'adverb and so on structures. Dhundhari words likewise have an arched structure on the reason of orientation (gender), tense, individual, number, etc. Dhundhari can likewise have an alternate type of words as in Hindi. A brief portrayal of thing (Noun), action word (verb), descriptor (adjective), and modifier (adverb) are shown here of Dhundhari language.

3.1 Noun

The meaning of a noun is the name. The name of any object, person, place, incident, or material is known as a noun.

Nouns in Dhundhari end with a sound of vowels and according to the vowels sound they end with, we can categorize the gender of the word. When a word ends with आं and ई vowel sound then the word is known as feminine and when the word ends with आ, अ, ओ, or ऊ vowels sound then the word is known as masculine. In a few cases, an exception is found as some feminine words end with अ vowel sound. For example, पोसाक is a feminine word that ends with a अ vowel sound. When a masculine and singular noun ends with a ओ vowel sound then it converts to आ vowel sound in a plural form and case of a feminine noun, when singular words end with ई then it converts to आं vowel sound in a plural form.

Those Masculine noun words which end with अ and ऊ vowels sound remain the same in a plural form.

For example, चाटू and मोट्यार are masculine nouns that remain in the form of चाटू and मोट्यार, when we convert these to plural form.

3.2 Adjective

When words describe the specialty or characteristics of a Noun or a Pronoun then those words are known as an adjective. In Dhundhari, masculine adjectives ending with a ओ vowel sound are singular adjectives and these adjectives convert to plural with the end of आ vowel sound form. For example, चोखो is a singular form of a masculine adjective which converts into चोखा as a plural form. In the case of feminine, adjectives end with ई

vowel sound in both singular and plural form. For example, सोवणी is both singular and plural form. Some words of adjectives those end with अ vowel sound remain the same in a plural. For example, चतर is both the plural and singular form of the adjective.

3.3 Verb

When words or groups of words describe an action of a Noun or a Pronoun then these types of words are known as verbs. In Dhundhari, verbs form depends on tenses, and verb changes according to the person, number, gender, and case.27 forms of verbs can be formed by a single verb in Dhundhari. For example, 23 forms of verbs are formed by the verb आ. Forms of आ verb are following:

आबो,आबा,आज्यो,आतो,आती,आता,आवै,आवो,आवां,आवूंलीं,आवूंलों,आवूंलां,आवैली,आवैलो,आवैलां,आयो,आया,आयी,आ'र , आवूं,आवांलां, आवोला, आणो, आबाळी, आबाळो, आयेड़ी, आयेड़ो

Case: direct/oblique/vocative

Number: singular/plural

Gender: masculine/feminine

Person: First/second/third

Tense: present/past/future

Verbs end with बो in the direct case of Dhundhari whereas in the case of oblique, verbs end with बा and end with बो in a vocative case. For example, आबो is a verb of a direct case and it's written as आबा in the oblique case and आबो in the vocative case. When a verb has been done in the past then the verb ends with 'र. Here आ converts to आ 'र. When there is a doubt in performing a verb then आ converts to आतो in a masculine case, आती in a feminine case, and आता in a plural form. When there is another verb present in a sentence then आ converts to आणो. All the forms of a verb आ according to the gender, person, number, and tenses can be understood by the following tables:

Table 1. आ verb in Past Tense

	Masculine singular	**Masculine plural**	**Feminine singular**	**Feminine plural**
First person	आयो छो	आया छा	आयी छी	आया छा
Second person	आयो छो	आया छा	आयी छी	आया छा
Third person	आयो छो	आया छा	आयी छी	आयी छी

Table 2. आ verb in Present Tense

	Masculine singular	**Masculine plural**	**Feminine singular**	**Feminine plural**
First person	आवूं छूं	आवां छां	आवूं छूं	आवां छां
Second person	आवै छै	आवो छो	आवै छै	आवो छो
Third person	आवै छै	आवै छै	आवै छै	आवै छै

Table 3. आ verb in Future Tense

	Masculine singular	**Masculine plural**	**Feminine singular**	**Feminine plural**

First person	आवूंलो	आवांलां	आवूंली	आवांलां
Second person	आवैली	आवोला	आवैली	आवोला
Third person	आवैलो	आवैलां	आवैली	आवैली

When words do not change according to the object, gender, subject and tense then the words are known as indeclinable words. Conjunction, adverb, and postposition come under a category of indeclinable words.

The following table is a representation of the indeclinable words of Dhundhari.

Table 4. Indeclinable words

Category	Example
Conjunction	अर(और), ईंताणी(इसलिए), कै(कि), क्यूंकै(क्योंकि), खै-खै(या-या),पण(पर),नतर(नहीं तो), जदां(तब), नै-नै(ना-ना), जदांई(तभी)
Postposition	क(के), नै(को), सूं(से), ताणी(के लिए), को(का), की(की), पै(पर), मं(में)
Adverb	लगतमार(लगातार), नीड़ा (लगभग), माळै(अंदर)

3.4 Morphemes

Linguistic investigation is the logical examination of language with an emphasis on the properties and qualities of a language. Language structure is partitioned into phonetics, phonology, morphology, grammar, and pragmatics. As it has been discussed that Morphology explores the development of words in a language. A morpheme is the littlest significant unit of language structure with importance and can't be separated into more modest units. Since morphemes make up all words in the English language, learning morphemes opens the construction and significance inside words. This, thusly, helps with the educational experience.

Morphemes are the smallest grammatical unit. Morphemes are either free or bound and are utilized as prefixes, postfixes, roots, and bases in words. A free morpheme is an independent word, similar to "canine." "Canine" can't be broken into more modest morphemes without losing the word's importance. Bound morphemes can't remain without anyone else as words, for example, the - s in "pens."

Derivational and inflectional morphemes are bound morphemes. Root and base words are morphemes that structure the base or foundation of a word. A prefix morpheme joins to the front of a root or base morpheme, while a postfix will interface with the end. Inflectional morphemes are postfixes, and derivational morphemes can be prefixes or additions. At the point when free morphemes are consolidated, they structure compound words. Complex words are made by framing base or root morphemes with derivational morphemes

4. Proposed Algorithm:

A Rule-based approach is used for developing lemmatizer. For that more than 200 rules were created. First affix from given or selected word is removed .After that lemma according to removed affix is added. List of added lemmas according to removed suffixes are mentioned in Table-5:

Table 5. Removed Suffixes and added Lemmas

Removed suffix	Lemma
यां	◌ी
◌ा◌ं	◌ो
यों	ई / ◌ी
◌ाई	◌ो / ◌ी/◌ी

◌ो	◌ै/ ◌ो
◌ी	◌ै/ ◌ी
यो	◌ा/ ◌ी/ ◌ा
◌ा	◌ो
◌ों	◌ो
◌ोड़ो	◌ा
◌ित	◌ा

Table 6. Some Examples

Word	Removed affix	lemma	result
छोर्यां	यां	◌ी	छोरी
छोरां	◌ां	◌ो	छोरो
लुगायां	यां	ई	लुगाई
जुताई	◌ाई	◌ो	जोत

Algorithm

```
str← string of input words
P ← List of Prefixes
S ← List of Suffixes
W=str.split()
calculated Length of each suffix in s
for suffix in s do
L ← List of Length of each suffix
For suffix in s do
For word in w do
If     word[-max(L):] ==suffix:
Output     word [: -max(L)], suffix.
Elif   word [-(max(L)-1):] == suffix:
Output   word[:-(max(L)-1)], suffix.
Continuously search suffix in given word accordingly until
If Len(word[-max(L)-N:]) >= min(L), where N=0,1,2,3….
End for
End for
Output   combined List←all words after suffix removal and removed suffixes
(word, removed suffix)
```

```
Output   R ← All Input words with suffixes.
Eliminate all Input words with suffixes from w,
We get,
W-R= Z
Add combined List with Z,
We get,
Z+ Combined List = F
For prefix in p do
Output LP ← List of Length of each prefix in p.
For prefix in p do
For word in F do
If     word [0] [:max (LP)] == prefix:
Output     Replacement of word with word [max (LP):] in F.
Continuously search prefix in given word accordingly,
If Len (word [0] [:max (LP)-N]) >= min (LP): where, N=0,1,2,3…
End for
End for
Output F
For stem in F do
If Len(stem) >0:
If stem [1] ==”ो”:
S1← combine”ो” with stem [0] at the end.
Output S1
Add lemma according to stem [1] of a stem [0]
Output Lemma ←List of S1
Else:
Output stem.
End for
```

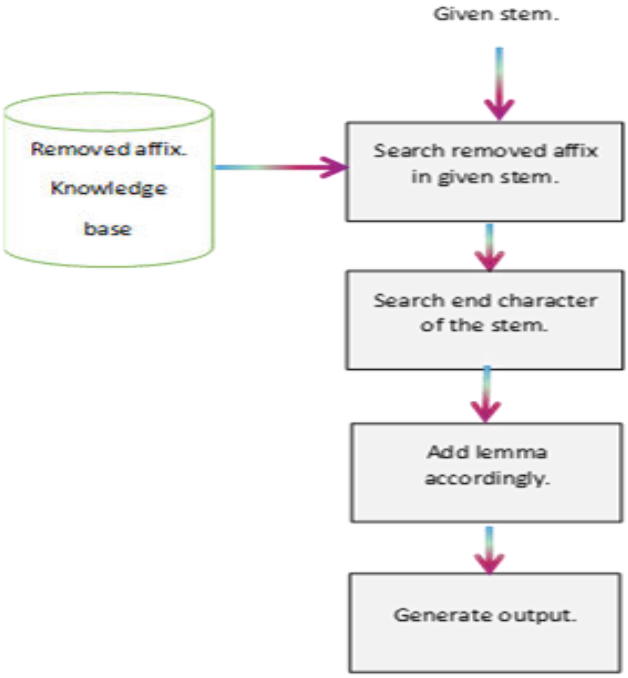

Fig.1. Workflow of Lemmatization for Dhundhari

5. Evaluation And Results

The system was evaluated using the standard accuracy score (equation 1). The system was tested for 2000 words. Among these 2000 words, 1000 were inflectional and the other 1000 words were derivational. In the case of inflectional words, the system was able to correctly identify 992 words which gave an accuracy of 99.2%. In the case of derivational words, the system was able to correctly identify 990 words which gave an accuracy of 99%. overall, out of the 2000 words the system was able identify 1982 words. This gave an overall accuracy of 99.1%. Table -7 summarizes this study.

$$Accuracy = \frac{Correct\ output}{words} \qquad (1)$$

Table 7. Representation of the Accuracy of the Results

	Total words	Correct output	Incorrect output	Accurate percentage
Inflectional	1000	992	8	99.2
Derivational	1000	990	10	99
Combined	2000	1982	18	99.1

6. CONCLUSION

For analysing the morphology of any language, the processes of Stemming and Lemmatization are to be applied and explored. Stemming is used to extract root word from any word of the language. Sometimes stemming process might suffer the problem of over stemming and under stemming. The problem of over stemming can be resolved by lemmatization. In this paper rule-based algorithm of lemmatizer for Dhundhar is developed. Dhundhari is found to be the extremely low resource and very rich morphological language. First time Lemmatizer for Dhundhari is developed.While developing Lemmatizing algorithm for Dhundhari 200 rules were created. The Inflectional and derivational lemmatizer were evaluated separately. The inflectional accuracy was

99.2% and derivational accuracy was 99%. Overall combined accuracy (Inflectional+ Derivational) was 99.1%. The system was also evaluated for accuracy and user acceptability.

A greater variety of data can be tried for evaluation and measurement of the effectiveness of developed lemmatizer. Some limitations of the developed lemmatizer were also observed. Since Dhundhari is a dialect of Hindi, in certain cases the words of Dhundhari and Hindi are similar which caused the conflict in lemmatizing rules. Apart from that some words are creating problem. For example, getting of "छोरी" as an output of input word "छोरी" is correct, but getting of "मारवाड़ी" as an output of input word "मारवाड़ी". Here output should be "मारवाड़". These are areas those need refinement and will be addressed in future versions.

Biography of Authors

Varda is Research scholar in Computer Science and Engineering, at Banasthali Vidyapith, Banasthali, Rajasthan India. She is also working as faculty in the department of Computer Science and Engineering, Manipal University Jaipur, Rajasthan India.

She has completed her Master degree from Banasthali University, Rajasthan, India.

Her research interests are in the areas of **Machine Learning and Natural language Processing**. She had published research articles in reputed National and International journals of Computer sciences. She has published one **patent** on Machine translation of Dhundhari Language.

Nisheeth Joshi is an Associate Professor at Banasthali University, India. He has done his PhD in the area of Natural Language Processing. Being involved in teaching and research for over 10 years, he has developed the art of explaining even the most complicated topics in a straight forward and easily understandable fashion. He also has vast experience of handling large scale research projects. This has helped him in developing practical insights of complex AI systems. He has authored several papers in international journals and conferences on various topics that includes Machine Translation Evaluation, POS Tagging, Morphology, Name Entity Translation, Text Simplification etc.

REFERENCES

1. Al-Shammari, E., & Lin, J. (2008, July). A novel Arabic lemmatization algorithm. In *Proceedings of the second workshop on Analytics for noisy unstructured text data* (pp. 113-118).
2. Bergmanis, T., & Goldwater, S. (2018, June). Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1391-1400).
3. Dhar, P., Bisazza, A., & van Noord, G. (2022, June). Evaluating Pre-training Objectives for Low-Resource Translation into Morphologically Rich Languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 4933-4943).
4. Hafeez, R., Anwar, M. W., Jamal, M. H., Fatima, T., Espinosa, J. C. M., López, L. A. D. & Ashraf, I. (2023). Contextual Urdu Lemmatization Using Recurrent Neural Network Models. *Mathematics*, *11*(2), 435.
5. Ingólfsdóttir, S. L., Loftsson, H., Daðason, J. F., & Bjarnadóttir, K. (2019). Nefnir: A high accuracy lemmatizer for Icelandic. *arXiv preprint arXiv:1907.11907*.
6. Jongejan, B., & Dalianis, H. (2009, August). Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 145-153).
7. Kanerva, J., Ginter, F., & Salakoski, T. (2021). Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks. *Natural Language Engineering*, *27*(5), 545-574.
8. Lo, C. K., Cherry, C., Foster, G., Stewart, D., Islam, R., Kazantseva, A., & Kuhn, R. (2016, August). NRC Russian-English machine translation system for WMT 2016. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (pp. 326-332).
9. Manjavacas, E., Kádár, Á., & Kestemont, M. (2019). Improving lemmatization of non-standard languages with joint learning. *arXiv preprint arXiv:1903.06939*.
10. Paul, S., Tandon, M., Joshi, N., & Mathur, I. (2013, July). Design of a rule based Hindi lemmatizer. In *Proceedings of Third International Workshop on Artificial Intelligence, Soft Computing and Applications, Chennai, India* (Vol. 2, pp. 67-74).
11. Perera, P., & Witte, R. (2005, October). A self-learning context-aware lemmatizer for German. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (pp. 636-643).
12. Plisson, J., Lavrac, N., & Mladenic, D. (2004, October). A rule based approach to word lemmatization. In Proceedings of IS (Vol. 3, pp. 83-86).
13. Rodrigues, R., Gonçalo Oliveira, H., & Gomes, P. (2014). LemPORT: a high-accuracy cross-platform lemmatizer for portuguese. In *3rd Symposium on Languages, Applications and Technologies*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
14. Salih, H. M. O., Devi, M. R., Ali, D. A. A., & Elmula, E. M. A. F. (2023). A Computational Analysis of Arabic Noun Morphology. *International Journal of Linguistics, Literature and Translation*, *6*(3), 52-62.
15. Shaukat, S., Asad, M., & Akram, A. (2023). Developing an Urdu Lemmatizer Using a Dictionary-Based Lookup Approach. *Applied Sciences*, *13*(8), 5103.

16. Straková, J., Straka, M., & Hajic, J. (2014, June). Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 13-18).

17. Yeong, Y. L., Tan, T. P., & Mohammad, S. K. (2016). Using dictionary and lemmatizer to improve low resource English-Malay statistical machine translation system. *Procedia Computer Science*, *81*, 243-249.

18. Zhang, R., & Sumita, E. (2007, June). Boosting statistical machine translation by lemmatization and linear interpolation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions (pp. 181-184).