Surface Realization

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Second Year B.Tech.

by

Avi Chawla, Ayush Sharma, Shreyansh Singh

Under the guidance of

Dr. Anil Kumar Singh

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2018

Dedicated to

My parents and teachers

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date: 02-05-2018

Avi Chawla, Ayush Sharma, Shreyansh Singh
B.Tech. Students
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Surface Realization**" being submitted by **Avi Chawla (Roll No. 16075014), Ayush Sharma (Roll No. 16075016), Shreyansh Singh (Roll No. 16075052)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: 02-05-2018

Dr. Anil Kumar Singh
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

We would like to express our sincere gratitude to Dr. Anil Kumar Singh for his constant guidance throughout the duration of the project. Our discussions with him, in which we discussed ideas related to our project, proved to be very fruitful. Also, we are thankful to our seniors for their help in overcoming the smaller difficulties we faced in our way. Last but not the least, we would like to thank our parents for their due consideration to our busy schedule, providing us constant motivation and support along with their unconditional love.

Place: IIT (BHU) Varanasi

Date: 02-05-2018 **Avi, Ayush, Shreyansh**

Abstract

This is the report of our exploratory project done in Even Semester, 2017-18. The problem statement was proposed in Surface Realization Shared Task, ACL 2018, Melbourne, Australia. The problem is stated as "Convert genuine UD structures from which word order information has been removed and the tokens have been lemmatized into their correct sentential form". The dataset of the shared task comprises of two sets of files, a .conll file containing the UD structures of sentences, and a text file containing the ordered sentences. We have majorly used statistical and probabilistic approaches to solve the problem.

Contents

List of Figures **ix**

List of Tables **ix**

1 Introduction **2**

1.1 Overview 2

1.2 Motivation of the Research Work 4

1.3 Organisation of the Report 4

2 Sub-Problem 1: Reinflection **5**

2.1 Approach 1 5

2.2 Approach 2 6

3 Sub-Problem 2: Word Ordering **8**

3.1 Approach 1 8

3.2 Approach 2 8

4 Results and Future Work **10**

4.1 Results 10

4.2 Future Work 11

Bibliography **12**

List of Figures

1.1 Sample Dependency Tree 2

1.2 PoS Tag Sequence 3

2.1 Architecture of the model 6

List of Tables

4.1 Results for Sub-Problem 1 10
4.2 Results for Sub-Problem 2 11

Chapter 1

Introduction

1.1 Overview

Universal Dependency (UD) [1] structure is a tree representation of the dependency relations between words in a sentence of any language. Made using the UD framework, the structure of the tree is determined by the relation between a word and its dependents. Each node of this tree holds the PoS tag and morphological information as found in the original annotations of the word corresponding to that node.

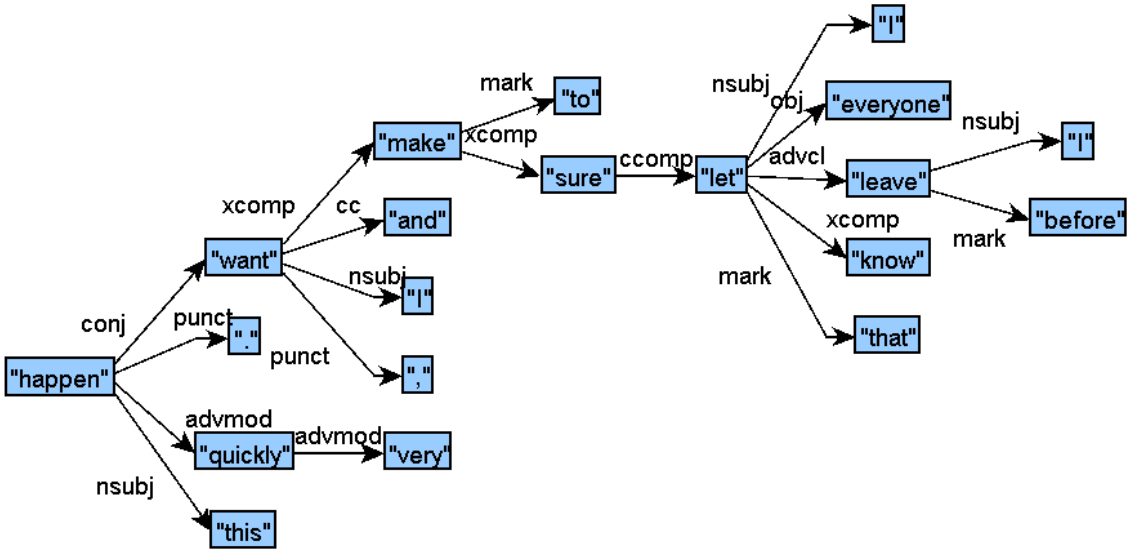

Figure 1.1 Sample Dependency Tree

PoS tag is the part of speech tag assigned to a word, based on its definition and context. The context could be its relationship with adjacent words in a phrase or sentence. Example of a few PoS tags are NN (Noun), JJ (Adjective), VB (Verb) and DT (Determiner).

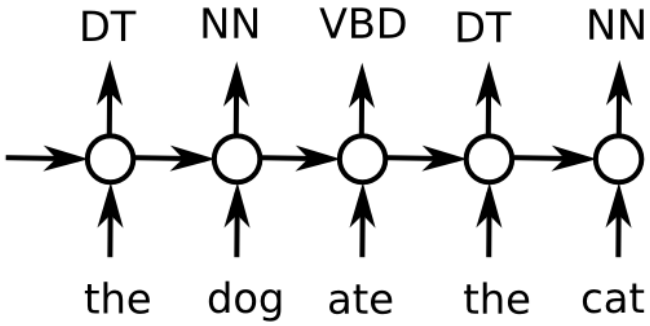

Figure 1.2 PoS Tag Sequence

The morphological information of a word includes the information gained from the formation of the word and its relationship with other words. Morphological information includes gender, animacy, number, mood, tense etc.

In this problem, we are given -

1. Unordered dependency trees with lemmatized nodes
2. The nodes hold PoS tags and morphological information as found in the original annotations
3. The corresponding ordered sentences

Our objective is to reinflect the lemmatized words and then determine the correct word order for each sentence consisting of these inflected words in an incorrect order.

1.2 Motivation of the Research Work

We wanted to work on a new and challenging task in our desired field of Natural Language Processing. For our exploratory project, we asked our Professor-in-charge to give us a chance to work on such a problem. On receiving the problem statement and on knowing that it has been proposed in the shared task of an A* conference (ACL), we were very excited to take up the project. Since it was a relatively new problem, we were not able to find many resources for reference. Still, the very intriguing nature of the problem statement, along with constant support of our Professor-in-charge and seniors, kept us motivated throughout this duration.

1.3 Organisation of the Report

We have divided our problem into two sub-problems, sub-problem 1 and sub-problem 2. All the approaches we have tried for sub-problem 1 and sub-problem 2 have been described in Chapter 2 and Chapter 3 respectively. The results of both the sub-problems have been presented in Chapter 4. Chapter 4 also includes some of the approaches which can be experimented with to solve the problem.

Chapter 2

Sub-Problem 1: Reinflection

In the given UD structure, the words are given in lemmatized form. Before proceeding to determine the correct order of words, these lemmatized words must be reinflected to convert them to their correct form. For the task of reinflection, we implemented two approaches- a string alignment based approach [2] and an LSTM [3] based encoder-decoder model [4].

2.1 Approach 1

The string alignment based approach only accounts for the prefix and suffix changes in inflecting a lemmatized word to its correct form. As the first step, the model aligns input and output training examples using Levenshtein distance. We then, divide the words into Prefix, Stem and Suffix.

Then, we extract the prefix-changing rules from the Prefix part and the suffix-changing rules from the Stem+Suffix part. This process is done for each UPoS (Universal part of Speech tag).

For prediction, we apply the longest suffix rule that applies to a lemma form to

be inflected for the given target UPoS. After that, the most frequently seen prefix-changing rule is applied. For unseen strings, the lemmatized word itself is assumed to be the the reinflected word.

2.2 Approach 2

This approach models the problem as one of generating a character sequence, character-by-character using an encoder-decoder LSTM model.

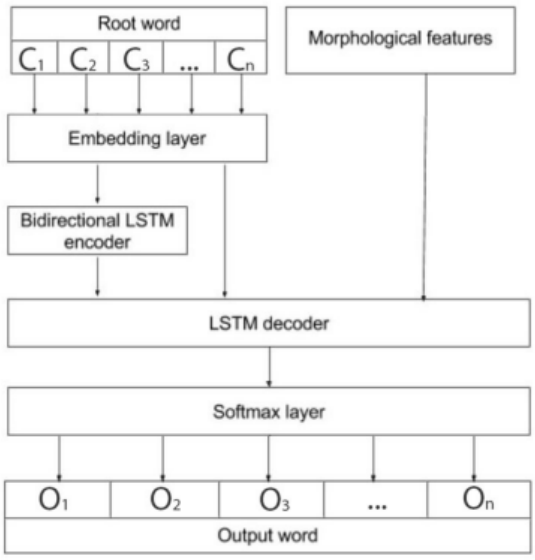

Figure 2.1 Architecture of the model

A character embedding is made for each character of the root word and this embedding is fed into a bidirectional LSTM encoder. The output of this encoder, along with the root word embedding, is fed into a single layer LSTM decoder. A softmax layer is then used to predict the character that must occur at each character position of the target word.

After training the model, we reinflect the words in the UD structure. Then, we

make unordered sentences using the modified UD structure. The ordered sentences were already given as a part of the training data. This corpus of unordered and ordered sentences forms the training data for our next sub-problem.

Chapter 3

Sub-Problem 2: Word Ordering

3.1 Approach 1

This is Best PoS Tag Sequence Prediction Approach.

This approach utilizes the correlations between PoS tags to find how probable one tag is to appear after another tag in a sequence. To find these correlations, we first train our model on proper sentences to learn these probabilities.

Now to find the correct sequence of words, we follow the steps below:

- We enumerate over each possible permutation of the given set of jumbled words. Our aim here is to find that very sequence which corresponds to the maximum score of its PoS sequence.
- A score is calculated at each iteration and at last the sentence whose PoS sequence gave the maximum score is chosen.

The PoS Tagger used here was Spacy's PoS Tagger [5].

3.2 Approach 2

This is a Language Modeling (LM) based approach. For this, we use the SRILM toolkit [6].

3.2. Approach 2

First, we generate a vocabulary file from the corpus of ordered sentences. This vocab file, along with the ordered sentence data, is used to generate a .lm file. This file contains the probability scores of the associated n-grams (till trigrams).

To generate the correct sequence, we use the following algorithm:

- For a current sequence of length n, we choose the word from the list of remaining words, which gives the highest score for the resulting sequence of length n+1.
- Repeat the process till there are no words in the list of remaining words, i.e., n = length of the sentence.

The above procedure is repeated considering every possible word as the first word. The word with the largest score of its associated sentence is selected as the first word and the corresponding sentence is the predicted sentence.

Improvement 1: An improvement over the above mentioned method is to select the best 4-gram to be the starting sequence of the sentence. For determining rest of the sequence, the procedure is same as mentioned in the method stated above.

Improvement 2: A further modification to the above mentioned approach is to make different combinations of 4-grams, tri-grams and bi-grams, and select the best sequence for these n-grams which gives the highest score.

Chapter 4

Results and Future Work

4.1 Results

To evaluate the performance of our reinflection models, we use Accuracy as the evaluation metric.

Approach	Accuracy (in %)
String Alignment Based Approach	93.2
LSTM based encoder-decoder model	95.8

Table 4.1 Results for Sub-Problem 1

The word structure based approach misses out on examples that do not inflect words based on only prefix or suffix changes (eg. man to men). This problem is countered in the second approach as we predict each character of the inflected form as a separate output unit.

However, both the approaches perform badly when for the same morphological features, the inflected form for two different words vary greatly in structural changes.

To evaluate the performance of our word reordering models, we use BLEU Score [7] as the evaluation metric.

Approach	BLEU Score
Best PoS Tag Sequence Prediction Approach	1.9
LM Based Approach	4.1
4-gram improvement of LM Based Approach	21.3
Variable N-gram improvement-2 of LM Based Approach	30.9

Table 4.2 Results for Sub-Problem 2

The best PoS tag sequence prediction approach performs poorly because of the following reasons-

- A PoS tag sequence with a high score may not map to a meaningful sentence.
- There may be words with the same PoS tags in a sentence.

The LM based approach is better than the PoS tag sequence based approach because for a sentence to be meaningful, the relative ordering of its words is of prime importance, not the relative ordering of its PoS tags.
Since in this method, we determine the start of sentence word by word, we may encounter bigrams which are meaningful as a part of the sentence, but are not suitable to begin the sentence. To overcome this, we select the best 4-gram sequence as the start of the sentence. Alternatively, we can also make different combinations of variable length n-grams and select the one with the highest score.

4.2 Future Work

Till now, we have majorly worked with statistical and probabilistic techniques to tackle the problem. We wish to experiment with deep learning based approaches in the future. Since a dependency tree can be interpreted as a graph, using graph matching and searching techniques is another dimension we can explore.

Bibliography

[1] "Universal dependencies." [Online]. Available: http://universaldependencies.org

[2] R. Cotterell, C. Kirov, J. Sylak-Glassman, G. Walther, E. Vylomova, P. Xia, M. Faruqui, S. Kübler, D. Yarowsky, J. Eisner, and M. Hulden, "Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages," *CoRR*, vol. abs/1706.09031, 2017. [Online]. Available: http://arxiv.org/abs/1706.09031

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[4] A. Sudhakar and A. K. Singh, "Experiments on morphological reinflection: Conll-2017 shared task," in *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*. Association for Computational Linguistics, 2017, pp. 71–78. [Online]. Available: http://www.aclweb.org/anthology/K17-2007

[5] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[6] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP 2002*, 2002, pp. 901–904.

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135