Project Report on Sentence Simplification using Clause Identification
Natural Language Processing (CSE 472)
Team Linguists
Aman Sharma (2018201084)
Shreya Upadhyay(2018201091)

Abstract

After extensive research on Machine Translation and its several implementations, translation of complex sentences still remains a complicated process, which poses a road block in achieving accurate results. To enhance the preformance of translation of complex sentences, it is therefore, very necessary to focus on sentence simplification as a pre-processing step. Complex sentences constitute many phrases and clauses and our major task focusses on identification of the clauses in complex sentences. The report proposes a rule-based approach for clause identification and clause boundary detection using dependency parsing. Also, using the same approach, we suggest a strategy for breaking the complex sentence into simpler sentences.

1 Introduction

The current machine translation system suffers a set back due to the presence of complex sentences in the database. Due to the absence of any tool that converts complex sentences (multiple clauses being an indicator) to simpler ones, a translator has to manually do these types of translations, which is an expensive operation. This projects aims to bridge that gap using an approach called dependency parsing.
Dependency parsing is a category of sentence parsing that determines the roles of the words in a sentence in relation to other words regardless of their syntactic arrangement. With the evolution of universal dependencies, the syntactico-semantic relations between words can be classified multi-lingually, which is very useful for machine translation. By identifying the relations between the main verb and coordinate/subordinate verbs, we can easily classify the clauses, identify the possible relations between the verbs in the clauses and determine the most optimal clause boundary in that sentence based on word order.

2 Motivation

Long sentences with complex syntactic structure and long-distance dependencies can be troublesome for translation systems. Even though systems that use syntax trees or other syntactic constraints on the source or target side can theoretically reduce the impact of this problem however in practise simple models outperform them [1]. Intuitively, shorter sentences are easier to translate as has been pointed out in the context of both traditional [2][3][4] and neural [5] translation systems . Even though the introduction of the attention mechanism for neural machine translation by [5] mitigates the effects of long input sentences, we believe there is still room for improvement in dealing with long and complex inputs.

Current translation systems have no notion of the relative importance of source tokens in long input sentences. However, many such sentences could be simplified by removing information that is not crucial to retain the central sentence meaning. For example, the additional information provided by relative clauses interrupts the fluency of the main clause for the purpose of translation, while removing it would turn the higher-order structure as modelled in syntactic language models [6] or dependency language models [7] into local phenomena. Additional, non-central information can

also occur at the end of a sentence in the form of adverbials or coordinations which can make reordering decisions more difficult. For example, long range verb reordering in English to German translation may fail [8] or only be possible with low-scoring derivations.

Thus we need to incorporate an additional pre-processing step before feeding data into the machine translators that help to bridge the gap in the performance of translators. If complex sentences can be converted into simpler sentences, translation accuracy would positively boost up.

3 Literature Review

Since we are dealing with several sub-tasks, out literature review is divided into three sections: Machine Translation, Clause bouandary detection and Dependency Parsing.

3.1 Machine Translation

Since pre-processing affects the performance of machine translation, several approaches have been proposed in the past for sentence simplification. [8] describe an early approach to skeleton-based translation, which decomposes input sentences into syntactically meaningful chunks. The central part of the sentence is identified and remains unaltered while other parts of the sentence are simplified. This process produces a set of partial, potentially overlapping translations which are recombined to form the final translation.

A different approach to dealing with long-range dependencies are dependency language models [9] [10] which can score non-adjacent parts of a translation hypothesis and mix terminal and non-terminal symbols. In contrast, skeleton-based translation deals with these dependencies in the input and does not rely on potentially noisy dependency structures built up during decoding.

Hybrid machine translation tools such as [11] use a tree-like structure for simplification of the semantic structure in order to ease machine translation, a task strongly required for machine translation of complex sentences. In the vein of syntactic parsing for machine translation, Arabic to English MT have performed both morphological as well as syntactic preprocessing tools [12], [13], [14], which are useful to determine the sentence structure and reordering needed in cases of complex sentences. Similar approaches have been tried from Hindi to English by [15], while [16] use UNL in order to represent the relation of different lexical items, for an interlingua based MT system. For more data rich language pairs, statistical methods exist for sentence simplification before translation [17].

3.2 Clause Boundary Detection

Another important task in preprocessing for machine translation that has not been researched in the context of downstream tasks is clause boundary detection. [18] provides a survey of predicting clause boundaries, while [19] is a rule based method for clause boundary detection. The latter method is a pipeline that uses phrase structure trees in order to determine the clauses. The approach proposed by us employs the use of universal dependencies for clause detection. We are using this approach since ours is a very task-oriented approach that does not involve any word re-ordering and hence does not focus on word-order.

3.3 Dependency Parsing

The proposed approach uses universal dependencies for the reason that they capture common syntactic and semantic relations between words multi-lingually. [20] is the Universal Dependencies Project(v1) which is a multilingual treebank collection.

4 Experiments and Ideas

4.1 Experimental Analysis

The following experimental analysis was done on Google Translate, for translation of complex English sentences to Hindi.

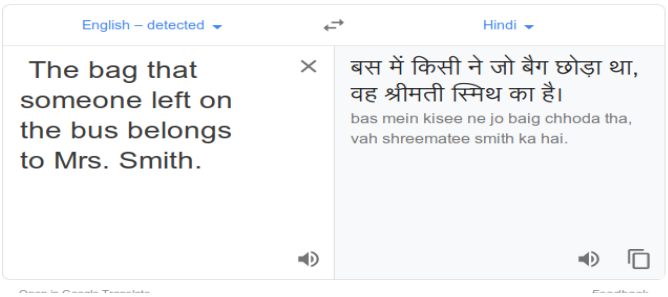

Figure 1

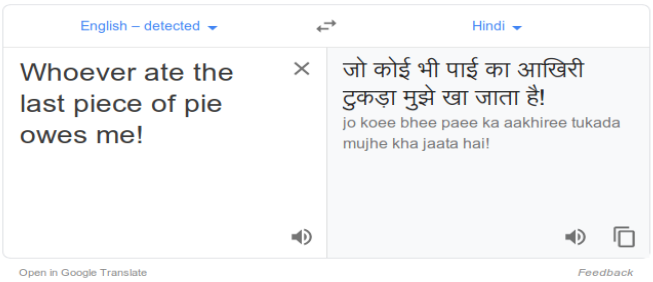

Figure 2.a

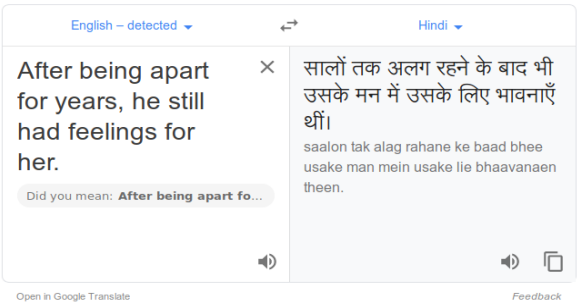

Figure 2.b

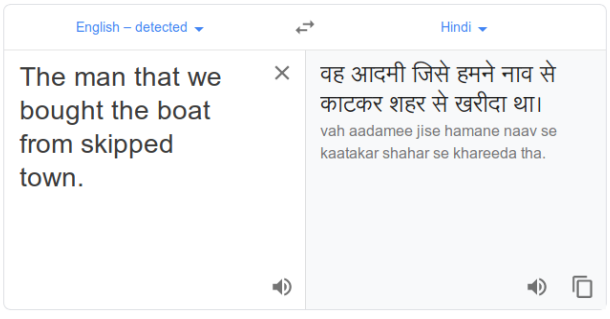

Figure 2.c

Figure 1 is an example of a simple sentence first, which is translated correctly to Hindi. However, Figure 2 shows various examples where the complexity in the sentences (presence of clauses), makes the translation rather inaccurate. In figure 2.a, the translation is bizarre due to the incorrect association of subject and object. In figure 2.b, the translation does justice to the sentence however usake can be ambiguously related to either the boy first and then the girl or the other way around. In figure 2.c, the presence of ‘that’ has created a havoc in translation as the dependencies have been disturbed.

From these translations, we concluded that it is not just the word orderings that matter in the translation but also the association of words with the subjects that is of central importance. If the verbs are incorrectly associated with their subjects, the sentence assumes a completely different meaning, far away from the intended one. Also, clauses play an important role in disturbing the subject-verb dependencies during translation.

Thus, dependency resolution seemed an important approach to solve this relation ambiguity in sentences containing more than one clause. Therefore, we resorted to dependency parsing using universal dependencies to identify sentences with more than two clauses and break them into simpler sentences.

4.2 Proposed Idea

The methodology used consists of three main steps : (1) Setting the criteria for a clause, (2) Algorithm for segmentation of clause and (3) Criteria for marking clause boundaries.

4.2.1 The Criteria for clause

The basic criterion for deciding whether or not a given segment of language constitutes a clause is the presence of a verb phrase, either finite (was, goes) or non-finite (be, gone, going). A verb phrase can obviously have more than one verb. This happens, for example, when the main verb is preceded by auxiliaries. It does not happen when the main verb is preceded by other main verbs. The sentence *"They must have been working"* has one verb phrase, since working is preceded by auxiliaries. On the other hand, the sentence *"They want to work"* has two verb phrases, since want and work are both main verbs, belonging, therefore, to two different clauses.

In general, the one-verb-phrase-one-clause criterion seemed to work satisfactorily, as long as the clauses were correctly separated.

4.2.2 Segmentation of clause using Dependency Parsing

For this step, we utilize Part of Speech Tagging (POS Tagging) to analyze what could be the possible patterns of tags that give us the clauses in a sentence. The proposed algorithm is as follows:

1. Identify if the sentence has more than one clause for which VERB tag is used as an indicator. If the sentences' POS tag list consists of two ore more VERB tags, it is considered a sentence of interest.

2. Create a dependency tree for the sentence and convert it to conllu form, a universal dependencies format. Below is an illustration of the format:

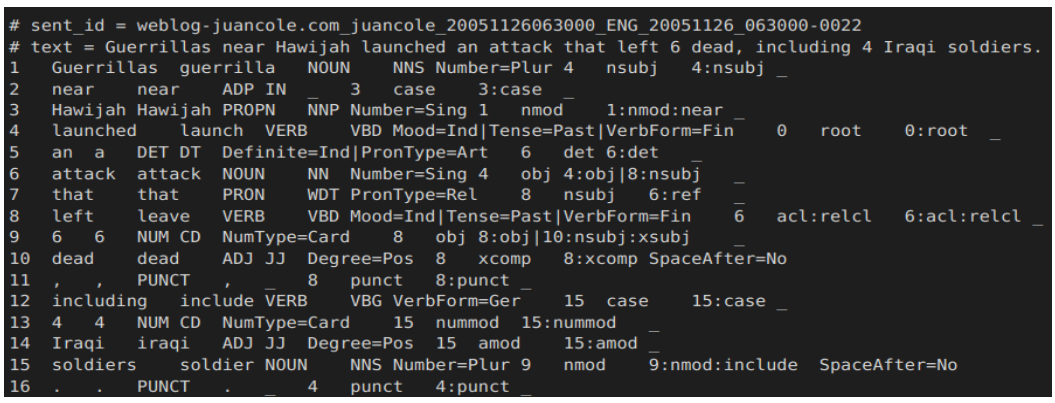

Figure 3

3. Loop throught the sentence's dependencies with realtion to NSUBJ. (Since each NSUBJ marks a new subsentence)

4. Loop through the NSUBJ dependencies. Out of the 37 dependencies in the universal dependencies, we consider the following dependencies: parataxis, ccomp, acl, acl:relcl, advcl, conj.

5. Create an ordered map of the words in the sentence. Start with adding the NSUBJ dependency into the map.

6. Till all the words in the sentence are not visited, identify the relationship (from the ones above) of words with the NSUBJ and add them to the ordered map.

At the end of the algorithm, we are left with clauses and the index of words that belong to that part of the sentence containing the clause.

4.2.3 Clause Boundary and Sentence Segmentation

In terms of sentence partitioning, a review of the literature suggests three ways in which a sentence can be segmented to the clause level: (1) starting with the first word in the sentence and processing it from left to right, word by word, until all the clauses are identified; (2) starting with formal indicators of subordination and coordination and proceeding until the end of the clause is found; (3) starting with the verb phrase, identifying the verb type and locating its subject and complements.

In our approach, clause boundary identification has been done by using linguistic rules which do not depends upon sentence boundaries. The following rules were taken into consideration :

Rule 1: If the current word is any relative clause marker (checked via the various relationships) and next word is any of the POS tags verb, pronoun, adjective, noun then the next word is marked as beginning of clause boundary.

Rule 2: If the current word is any verb auxiliary and next word is any symbol then current word is end of corresponding subordinate clause boundary.

When the respective words related to one clause are found using nearest noun approach, we segment the sentences into simpler sentences having one clause each.

5. Observation and Results

For example, let us consider the following sentence:

"They wanted to pick blueberries as a snack, but a bear growled at them from the berry patch."

Now this sentence is a complex sentence as it has two simple sentences joined by a coordinating cunjunction (CC tag) which is "but" .
Figure 3 is a table of test sentence where each token is represented in the 10 categories of the CoNLL-U format details of which can be found here: https://universaldependencies.org/format.html

For reference to POS tags please visit this link:
https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

```
['They', 'wanted', 'to', 'pick', 'blueberries', 'as', 'a', 'snack', ',', 'but', 'a', '
bear', 'growled', 'at', 'them', 'from', 'the', 'berry', 'patch', '.']
1       They    _       PRP     PRP     _       2       nsubj   _       _
2       wanted  _       VBD     VBD     _       0       root    _       _
3       to      _       TO      TO      _       4       mark    _       _
4       pick    _       VB      VB      _       2       xcomp   _       _
5       blueberries     _       NNS     NNS     _       4       dobj    _       _
6       as      _       IN      IN      _       8       case    _       _
7       a       _       DT      DT      _       8       det     _       _
8       snack   _       NN      NN      _       4       nmod    _       _
10      but     _       CC      CC      _       2       cc      _       _
11      a       _       DT      DT      _       12      det     _       _
12      bear    _       NN      NN      _       13      nsubj   _       _
13      growled _       VBD     VBD     _       2       conj    _       _
14      at      _       IN      IN      _       15      case    _       _
15      them    _       PRP     PRP     _       13      nmod    _       _
16      from    _       IN      IN      _       19      case    _       _
17      the     _       DT      DT      _       19      det     _       _
18      berry   _       JJ      JJ      _       19      amod    _       _
19      patch   _       NN      NN      _       13      nmod    _       _

Line violates:  ['10', 'but', '_', 'CC', 'CC', '_', '2', 'cc', '_', '_']
```

Figure 4

The next step, iteratively parses all the children starting from root of the dependency tree and checks for relations with the root. If the words have relations, mentioned previously, with the root, they are added to the sub-sentence of the clause under consideration.

```
['///root', 'They', 'wanted', 'to', 'pick', 'blueberries', 'as', 'a', 'snack', ',', 'b
ut', 'a', 'bear', 'growled', 'at', 'them', 'from', 'the', 'berry', 'patch']
Parent:  0   Children--->  [2]
Parent:  2   Children--->  [1, 4, 10, 13]
Parent:  1   Children--->  []
Parent:  4   Children--->  [3, 5, 8]
Parent:  3   Children--->  []
Parent:  5   Children--->  []
Parent:  8   Children--->  [6, 7]
Parent:  6   Children--->  []
Parent:  7   Children--->  []
Parent:  10   Children--->  []
Parent:  13   Children--->  [12, 15, 19]
Parent:  12   Children--->  [11]
Parent:  11   Children--->  []
Parent:  15   Children--->  [14]
Parent:  14   Children--->  []
Parent:  19   Children--->  [16, 17, 18]
Parent:  16   Children--->  []
Parent:  17   Children--->  []
Parent:  18   Children--->  []
```

Figure 5

Figure 4 shows the iterative functioning of the algorithm to determine the clause boundaries and the words belonging to each sub-sentence of a complex sentence.
After running the algorithm for sentence creation for each clause, we obtain the following result:

```
Input Sentence: They wanted to pick blueberries as a snack, but a bear growled at them from the berry patch.
Clause1: They wanted to pick blueberries as a snack, but
Clause2: a bear growled at them from the berry patch
```

As we can see, the algorithm works fairly well for sentences with multiple clauses and sentence segmentation is also correct. Thus we can conclude that the algorithm works well for a fair length of a sentence.

Let us consider another example:
“Bell, a telecommunication company, which is based in Los Angeles, makes and distributes electronic, computer and building products.”

Our algorithm fails to mark clause boundaries for this sentence due to ambiguity in tags since it has clauses for which NSUBJ is not a valid POS tag (as in Figure 5). Therfore we need to add dependency rules for sentences with **CONJ** and **CC**, not just **NSUBJ**.

```
Enter a Sentence to test--> Bell, a telecommunication company, which is based in Los A
ngeles, makes and distributes electronic and computer building products.
1       Bell            _       NNP     NNP     _       14      nsubj   _       _
3       a               _       DT      DT      _       5       det     _       _
4       telecommunication               _       JJ      JJ      _       5       amod    _       _
5       company         _       NN      NN      _       1       appos   _       _
7       which           _       WDT     WDT     _       9       nsubjpass       _       _
8       is              _       VBZ     VBZ     _       9       auxpass _       _
9       based           _       VBN     VBN     _       1       acl:relcl       _       _
10      in              _       IN      IN      _       12      case    _       _
11      Los             _       NNP     NNP     _       12      compound        _       _
12      Angeles         _       NNP     NNP     _       9       nmod    _       _
14      makes           _       VBZ     VBZ     _       0       root    _       _
15      and             _       CC      CC      _       14      cc      _       _
16      distributes             _       VBZ     VBZ     _       14      conj    _       _
17      electronic              _       JJ      JJ      _       20      dep     _       _
18      and             _       CC      CC      _       20      cc      _       _
19      computer                _       NN      NN      _       20      conj    _       _
20      building                _       NN      NN      _       21      amod    _       _
21      products                _       NNS     NNS     _       14      dobj    _       _

--------------------------------------------------------------------------------
```

Figure 6

(Github Repository : https://github.com/shreyaUp/Sentence-Simplification)

6. Paper Presentation

The paper presentation is based on the paper : Clause Identification in English and IndianLanguages:A Survey, Misha Mittal, Abhilasha
Link: http://ijoes.vidyapublications.com/paper/Vol13/35-Vol13.pdf

7. References

[1] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, M. Turchi, **Findings of the 2015 workshop on statistical machine translation,** Proceedings of the 10th Workshop on Statistical Machine Translation (2015)

[2] B. Mellebeek, K. Owczarzak, D. Groves, J. van Genabith, A. Way, **A syntactic skeleton for statistical machine translation,** Proceedings of the 11th Conference of the European Association for Machine Translation) (2006)

[3] Xiong H., Xu W., Mi H., Liu Y., Liu Q., **Sub-sentence division for tree-based machine translation,** Proceedings of the Association for Computational Linguistics and Asian Federation of Natural Language Processing 2009 Conference Short Papers (August) (2009), pp. 137-140

[4] K. Sudoh, K. Duh, H. Tsukada, T. Hirao, M. Nagata, **Divide and translate : Improving long distance reordering in statistical machine translation,** Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics (2010), pp. 418-427

[5] D. Bahdanau, Cho K., Y. Bengio, **Neural machine translation by jointly learning to align and translate,** Proceedings of the International Conference on Learning Representations (2015), pp. 1-15

[6] A. Pauls, D. Klein, **Large-scale syntactic language modeling with treelets,** Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (July) (2012), pp. 959-968

[7] Shen L., Xu J., R. Weischedel, **A new string-to-dependency machine translation algorithm with a target dependency language Model,** Proceedings of the Association for Computational Linguistics-08: Human Language Technologies (2008), pp. 577-585

[8] F. Braune, A. Gojun, A. Fraser, **Long-distance reordering during search for hierarchical phrase-based SMT,** Proceedings of the European Association for Machine Translation-2012 (May) (2012), pp. 28-30
[9] Shen L., Xu J., R. Weischedel, **A new string-to-dependency machine translation algorithm with a target dependency language Model,** Proceedings of the Association for Computational Linguistics-08: Human Language Technologies (2008), pp. 577-585
[10] R. Sennrich, **Modelling and optimizing on syntactic N-grams for statistical machine translation,** Proceedings of the Transactions of the Association for Computational Linguistics, 3 (2015), pp. 169-182
[11] Sinha, R. M. K., and A. Jain., **AnglaHindi: an English to Hindi machine-aided translation system**, MT Summit IX, New Orleans, USA (2003): 494-497
[12] Narayan, Shashi, and Claire Gardent, **Hybrid simplification using deep semantics and machine translation,** Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2014.
[13] El Isbihani, Anas, et al, **Morpho-syntactic Arabic preprocessing for Arabic-to-English statistical machine translation,** Proceedings of the Workshop on Statistical Machine Translation. Association for Computational Linguistics, 2006.
[14] Habash, Nizar, **Syntactic preprocessing for statistical machine translation,** Proceedings of the 11th MT Summit 10 (2007).
[15] Habash, Nizar, and Fatiha Sadat, **Arabic preprocessing schemes for statistical machine translation,** Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006.
[16] Rao, Durgesh, et al, **A practical framework for syntactic transfer of compound-complex sentences for English-Hindi machine translation,** Proceedings of KBCS. Vol. 2000. 2000.
[17] Dave, Shachi, Jignashu Parikh, and Pushpak Bhattacharyya, **Interlingua-based EnglishHindi machine translation and language divergence,** Machine Translation 16.4 (2001): 251-304.
[18] Specia, Lucia, **Translating from complex to simplified sentences,** International Conference on Computational Processing of the Portuguese Language. Springer, Berlin, Heidelberg, 2010.
[19] Sharma, Sanjeev Kumar, **Clause Boundary Identification for Different Languages: A Survey,** International Journal of Computer Applications Information Technology 8.2 (2016): 152.
[20] Sacaleanu, Bogdan, Alice Marascu, and Charles Jochim, **Rule-based syntactic approach to claim boundary detection in complex sentences,** U.S. Patent No. 9,652,450. 16 May 2017.
[21] Nivre, Joakim, et al, **Universal Dependencies v1: A Multilingual Treebank Collection,** LREC. 2016.
[22] White, Aaron Steven, et al, **Universal decompositional semantics on universal dependencies,** Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.
[23] Zeman, Daniel, et al, **CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies,** Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (2018): 1-21.
[24] Schuster, Sebastian, and Christopher D. Manning, **Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks,** LREC. 2016.
[25] Tandon, Juhi, et al, **Conversion from paninian karakas to universal dependencies for hindi dependency treebank,** Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016). 2016.