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.

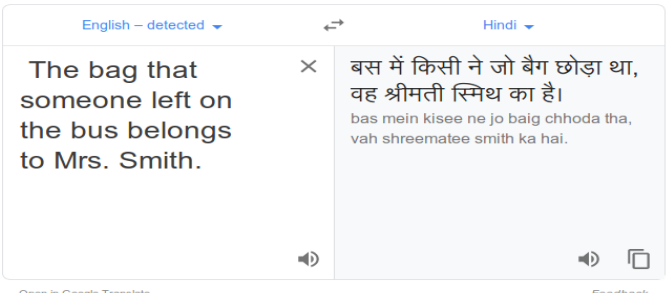


Figure 1

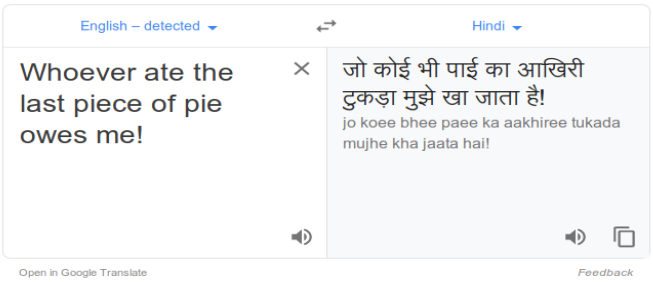


Figure 2.a

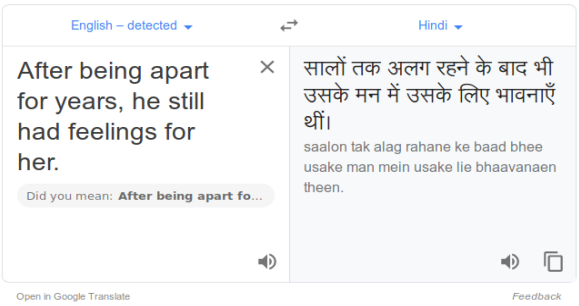


Figure 2.b

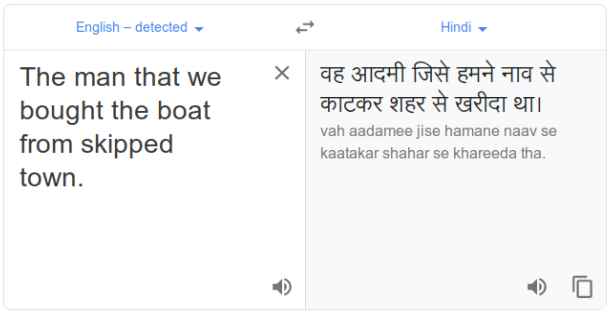


Figure 2.c

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 Clause Boundary Detection using Dependency Parsing

We utilize the dependencies obtained using dependency parsing (via the Stanford Dependency Parser). For the interpretation of the dependenices, we use the typed dependencies format that is as follows:

If we consider the following sentence: "John, who was the CEO of a company, played golf."

```
=========  Verbs in Sentence ==========
 Word :  was
 Tag:  VBD
 Word :  played
 Tag:  VBD
Sentence : " John, who was the CEO of a company, played golf. " is complex!
=====================================
```

Figure 3 . The sentence is correctly identified as complex due to presence of two verbs as shown.

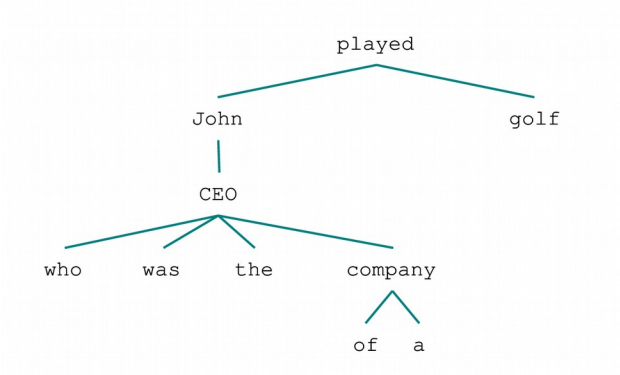


Figure 4. Dependency Parse Tree

```
============== TRIPLES =============
nsubj ('played', 'VBD') ('John', 'NNP')
acl:relcl ('John', 'NNP') ('CEO', 'NN')
nsubj ('CEO', 'NN') ('who', 'WP')
cop ('CEO', 'NN') ('was', 'VBD')
det ('CEO', 'NN') ('the', 'DT')
nmod ('CEO', 'NN') ('company', 'NN')
case ('company', 'NN') ('of', 'IN')
det ('company', 'NN') ('a', 'DT')
dobj ('played', 'VBD') ('golf', 'NN')
```

Figure 5. Triple Relation (Word1, Word2)

Since verbs are the main indicators of clauses, we use the sub-tree/ dependencies of verbs in order to identify the clause boundaries in the sentence. Thus all the children/dependencies of the verbs in the sentence form one clause. Thus we can obtain the clause boundaries for each of these clause. A clause can be identified from relation (in SD) which category is subject, e.g. nsubj, nsubjpass. (Stanford Dependency Manual). Therefore, we make use of this information to carry forward the task of finding clase boundaries.

1. Determine all the subject-verb dependencies : nsubj and nsubjpass.

2. For each of these subject-verb dependencies, find all the dependencies recursively, that have link to that dependency, except any dependency whose relation category is subject.

3. Find all the unique words in that list and sort them according to their appearance in the sentence.

4. The list gives us all the words that belong to that clause.

When we run our algorithm on the above example sentence. Two subject-verb dpendencies were identified, nsubj(played,John) and nsubj(CEO,who) respectively.

For each of these two sub-verb dependencies, we obtain all the dependencies. For the dependency, nsubj(played, John) we start with the dependencies of played which is dobj(played, golf). Similarly, for nsubj(CEO, who) we get the following dependency list : [cop(CEO, was), det(CEO, the), nmod(CEO, company) , acl:recl (John, CEO) , case(company, of) , det(company, the)]

So the clause boundaries for played include words [John-1, golf-12] and for was include words [John-1, the-5, CEO-6, of-7, a-8, company-9]

4.2.3 Sentence Segmentation

When we construct sub-sentences after sorting them based on their index, we obtain the following sentences:

1. John was the CEO of a company

2. John played golf

But not all sentences are as easily segmented as this one. There are several sentences that pose difficulties in rule generalisation for sentence splitting and we need to observe the outliers rather carefully. We are focussing on the following sentence categories for the analysis and rule generation:

1. Sentences with one independent clause (Eg. We played chess all evening).

2. Sentences with one independent clause and one dependent clause that is either

connected using relative pronouns (Eg. The book, which is now out of print, has all the information you need.) or subordinating conjunctions (Eg. She went to the school that my father went to.).

3. Sentences with two or more independent clauses that are connected by a conjunction(Eg. I will eat broccoli after I eat this cookie.).

4. Sentences with elided verbs and predicates.

Section 6 deals with the detailed analysis of these four categories.

5. Observation and Results

5.1 Identification of Simple and Complex sentences
The dataset used was English USD Dataset(in CoNLL-U format) to check if the algorithm for complexity detection works or not.

Number of Complex Sentences in Dataset = 3604
Number of Complex Sentences found by Algorithm = 3592
Accuracy = 99.66 %

5.2 Clause Boundary Identification
The analysis for this part was done manually by cross-referencing with the ClausIE: Clause-Based Open Information Extraction (https://d5gate.ag5.mpi-sb.mpg.de/ClausIEGate/ClausIEGate/). It lists all the possible clause forms and we manually check the ones concerned to our need.

Sentences with relative pronouns:
Example : I have a friend whose cat is annoying.
Grandma remembers a time when radio was popular. (adverbial)
12 sentences like the above two were taken and the algorithm correctly identified clause boundaries for 10 of them. Hence, accuracy for relative pronouns is 83.33%

Sentences with subordinating conjunctions:
Example : As Sherri blew out the candles atop her birthday cake, she caught her hair on fire.
15 such sentences were taken and manually checked to obtain 11 correct clause boundaries. The accuracy thus obtained is 73.33%

Sentences with Co-ordinating conjunctions:
Example : You can eat your cake with a spoon *or* fork.
14 such sentences were taken out of whic clause bondaries were correctly identified for 12 sentences thus giving us an accuracy of 85.714%

Sentences with direct speech:
Example : He said, “ Please forgive me.”
Direct speech has two clauses, a reporting clause, here He said and the one in the reported speech. For sentences like these, the algorithm returns the entire sentence as a single clause without breaking it into the reporing clause and discourse. For 10 such sentences, only 5 had correctly identified clause boundaries and the others made the whole sentence a clause.

6. Error Analysis

The main issue arises when we have to reconstruct sentences from these clauses to form meaningful sentences. Therefore, we need to add certain rules in order to make sure that the corner cases are handled and the split sentences are coherent and meaningful. For this purpose, we have divided our analysis into various categories.

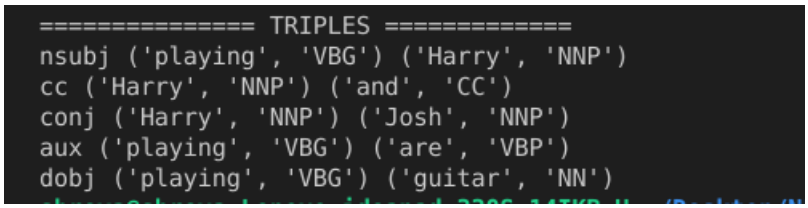
6.1 Sentences with conjucts

Consider the sentence, Harry and Josh are playing guitar. The sentence needs to be split as follows:

Harry is playing guitar.

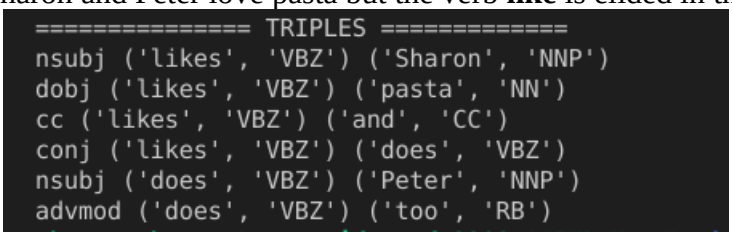
Josh is playing guitar.

However, due to only one nsubj dependency, we obtained a single sentence and incorrect clause boundaries. Below is the typed dependency representation:



On observing several similar sentences, we concluded that we can add the following rule:
Whenever there is a cc tag, the dependent in the conj dependency has the same object as the head of the dependency. Also, the verb's auxilliary should be made singular.

6.2 Elided Verbs
An elided verb phrase is one in which the non-finite verb has been left out (possibly because it can be inferred) . Verb phrase ellipsis is a common form of ellipsis and has certain rules. Only the non-finite verb can be elided. Ellipsis is introduced by an auxilliary verb (*be*, *can*, *do*, *don't*, *could*, *have*, *may*, *might*, *shall*, *should*, *will*, *won't*, *would*, etc.) or by the infinitive particle *to*.
Considering the sentence: Sharon loves pasta and Peter does, too.
We know that both Sharon and Peter love pasta but the verb **like** is elided in this case.



After examining several similar cases of verb ellipsis, we can add the following rule to our parser:
Transfer the object of the elided verb to the verb in conjuction with the elided verb. (here does)
This can help us obtain the following sentences:
Sharon likes pasta.
Peter likes pasta.

Another technique proposed in [26] present two methods for parsing to a Universal Dependencies graph representationthat explicitly encodes the elided material withadditional nodes and edges. This helps to reconstruct elided material from dependency trees with high accuracy when the parser correctly predicts the existence of a gap.

6.3 Sentences with clausal complement
For this case, we consider the sentence : I sold the car that I had just bought.

```
============== TRIPLES =============
nsubj ('sold', 'VBD') ('I', 'PRP')
dobj ('sold', 'VBD') ('car', 'NN')
det ('car', 'NN') ('the', 'DT')
ccomp ('sold', 'VBD') ('bought', 'VBN')
mark ('bought', 'VBN') ('that', 'IN')
nsubj ('bought', 'VBN') ('I', 'PRP')
aux ('bought', 'VBN') ('had', 'VBD')
advmod ('bought', 'VBN') ('just', 'RB')
```

After analyzing several similar sentences, we can add the following rule:

For ccomp relation, the dependant of the relation shares the same NN as the head of the relation. Therefore the object for both bought and sold would be ‘car’.

Split Sentences :

I sold the car.

The car I had just bought. (After adding the NN of the dobj)

6.4 Sentences with subordinating conjucts
For sentence “As Sherri blew out the candles atop her birthday cake, she caught her hair on fire .”

```
=============== TRIPLES =============
advcl ('caught', 'VBD') ('blew', 'VBD')
mark ('blew', 'VBD') ('As', 'IN')
nsubj ('blew', 'VBD') ('Sherri', 'NNP')
compound:prt ('blew', 'VBD') ('out', 'RP')
dobj ('blew', 'VBD') ('candles', 'NNS')
det ('candles', 'NNS') ('the', 'DT')
nmod ('blew', 'VBD') ('cake', 'NN')
case ('cake', 'NN') ('atop', 'IN')
nmod:poss ('cake', 'NN') ('her', 'PRP$')
compound ('cake', 'NN') ('birthday', 'NN')
nsubj ('caught', 'VBD') ('she', 'PRP')
dobj ('caught', 'VBD') ('hair', 'NN')
nmod:poss ('hair', 'NN') ('her', 'PRP$')
nmod ('caught', 'VBD') ('fire', 'NN')
case ('fire', 'NN') ('on', 'IN')
```

The advcl relation can be used to relate the pronoun she to Sherri using the nsubj dependency. This will transfer the subject to the second half of the sentence to give us the following split sentences:

Sherri blew out the candles atop her birthday cake.
Sherri caught her hair on fire.

Another example that we can take towards completion of split sentence is “This is the dog that was hit by that car”

```
============== TRIPLES =============
nsubj ('dog', 'NN') ('This', 'DT')
cop ('dog', 'NN') ('is', 'VBZ')
det ('dog', 'NN') ('the', 'DT')
acl:relcl ('dog', 'NN') ('hit', 'VBN')
nsubjpass ('hit', 'VBN') ('that', 'WDT')
auxpass ('hit', 'VBN') ('was', 'VBD')
nmod ('hit', 'VBN') ('car', 'NN')
case ('car', 'NN') ('by', 'IN')
det ('car', 'NN') ('that', 'DT')
```

For this sentence we can use the acl:relcl relation to give a subject to ‘hit’ which will be NN of acl:relcl i.e. dog. So the sentence would be segmented as:

This is the dog.
This dog was hit by that car.

(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.