Survey on Interpretable Semantic Textual Similarity, and its Applications

Abdo Ababor Abafogi

Abstract: Both semantic representation and related natural language processing(NLP) tasks has become more popular due to the introduction of distributional semantics. Semantic textual similarity (STS)is one of a task in NLP, it determinesthe similarity based onthe meanings of two shorttexts (sentences). Interpretable STS is the way of giving explanation to semantic similarity between short texts. Giving interpretation is indeedpossible tohuman, but, constructing computational modelsthat explain as human level is challenging. The interpretable STS task give output in natural way with a continuous value on the scale from [0, 5] that represents the strength of semantic relation between pair sentences, where 0 is no similarity and 5 is complete similarity. This paper review all available methods were used in interpretable STS computation, classify them, specifyan existing limitations, and finally give directions for future work. This paper is organized the survey into nine sections as follows: firstly introduction at glance, then chunking techniques and available tools, the next one is rule based approach, the fourth section focus on machine learning approach, after that about works done via neural network, and the finally hybrid approach concerned. Application of interpretable STS, conclusion and future direction is also part of this paper.

Keywords: Textual Semantic Similarity, Interpretable Textual Semantic Similarity, Application of Interpretable Textual Semantic Similarity, Deep Learning, Machine Learning, Rule based, Hybrid

I. INTRODUCTION

In this era, digitalization plays a vital rolelike transformingnecessary information in digital way. Social channels look up users' personal informationbefore launchingany product or tool[1]. Interpreting an information enhances effectiveness of a system performance. Interpretable STS is the way of giving meaning to semantic similarity between short texts. The final goal this survey would be to show an approach of interpreting semantic textual similarity and to showthe best one. Given the input (a pair of sentences), thendistinguisheach chunks in bothsentences, nextit computes similarity score for every possible pair of chunks based onthe given features. Finally align most related chunks between the two sentences, with a reasonwhy aligned. The relation interpreted asopposition, equivalence, similarity specificity, relatedness or unaligned (unrelated chunks)labeled with the similarity scorefromone to five. In interpretable STS dataset is crucial for both training and testing purpose.

Revised Manuscript Received on January 05, 2021.
* Correspondence Author

Abdo Ababor Abafogi*, B.Sc and M.Sc Information Technology, Jimma University

For this task Sem Eval 2015 and 2016 has been prepared dataset comprised pairs of sentences gatheredfrom image descriptions and headlines news. The images dataset consists of images with description whereas theheadlines dataset is collected from news headlines. Images dataset consist of 750 pairs of sentences used for training and 375 sentence pairs used for test.
On the other hand Headlines dataset consist of 756 pairs of sentences used for training and 375 sentence pairs used for test. Interpretable STS corpus has not existed in non-English language but recently Indonesian version is built [2].

Most importantly, recent work, show in what way the interpretable STS output used to produce descriptions automatically in NLP. Users achieved better result when additional clarification is given, in real applications [27].

All most in all works preprocessing is the first step to simplify the similarity calculation task. Many NLP tools are available for preprocessing, Stanford's NLP parser as wellas OpenNLP framework were usually preferred by many authors. Actually, all researchers performed some kind of text (input) operationlike tokenization, lowercasing, punctuation and stop word removal, lemmatization, parsing or part of speechtagging. Additionally named-entity recognition was used by many authors.

In order to identify chunks that the relationship between chunks is based on lexical selection, Abney [3] uses context-free grammar to describe the structure of chunks, providing a definition of a chunk from a linguistic perspective, which he hypothesizes is closer to how humans parse texts.

A chunk is a textual unit (a sequence) of adjacent wordsgrouped together basis on their part of speech tag thatindicatetheir internal relations [4], [6]. Based on this linguistic properties, chunking isparsing the sentence into a chunkbased sentence structureform.Many linguistic parser used to chunking the input sentences with some post processing [7].

To chunk the input sentences the authors of NeRoSim system [5] created a rule based chunks determination. Inspire system uses Answer Set Programming to determine chunk boundaries [10]. In ExB Themis systema default Open NLP chunker, is used[8].On other hand OpenNLP chunking tool output modified based on a rulesobserved fromthe dataset[7]as well asbased on their dependency [9], [14].In order to maximize chunk accuracy enormous of rules were discovered [13].Some of thoserules concerns how punctuations, conjunctions, and prepositions are handled [11], [12], [27]. Primarily concatenating two or more chunks (like preposition and noun phrase,noun phrases and conjunctions) and forming new chunks[27], [13], [14].

Another chunking tool was developed based on Conditional Random Fields (CRF) using both CoNLL-2000 shared task training data and test data. It generates shallow parsing features such as previous and next words from current word, and their POS tags [14].Having present notion of chunking and available tools for this purpose let's look at possible ways of chunk alignment score calculation and type prediction. Several approaches were proposed including rule based, machine learning (ML), neural network and hybrid approach.

II. RULE BASED

To this date, a few of the interpretable STS systems were rule-based that built on top of too much use of linguistic features and resources. The works conducted based on rule derived from observation of training dataset. One of well-known rule based system is NeRoSim [5] which depends on two methods, which is corpus and knowledge based. As corpus based NeRoSim used pre-trained Mikolov word representations [15] as knowledge based methods NeRoSim strictly lookup synonym, antonym and hypernyms features from WordNet. Chunks aligned twice to optimize alignment to calculate sentence similarity as in Stefanescu et al. [26, [5]. LexiM [16] is another rule based system purely based on lexical overlap (string or sub-string matches) contained 13 rules. Rev system extends LexiM by implementing the rules of LexiM and additional rules by manual data analysis of PoS categories and synonyms from the headlines training set. Rev system works with string distance similarity for lexical overlap, PoS match and semantic similarity (i.e. synonyms) based strategies [16]. Based on rule while aligning a given chunk pair, NeRoSim checks has 7 defined conditions. Moreover, a precedence of ruleswell definedfor all relation types and NOALIC relation assigned to a chunkas the last option.

Similarity score between the chunk pair using Mikolov word vectors. If a chunk to be mapped has no match, NOALIC assigned. For type EQUI 3 rules are applied by precedenceunconditionally, the rest rules are applied only if none of these conditions 1 to conditions 5 are satisfied. OPPO type assigned fora content word inchunkof sentence 1has an antonym in the chunk of sentence 2 [5].

If chunk X contains all content words of chunk Y plus some extra content words that are not verbs, X is a SPE of Y or vice-versa. If chunk X contains only one noun and chunk Y contains only one noun and if chunk X noun is hypernym of chunk Y noun, chunk Y is SPE of the chunk X or vice versa [5]. SIMI type assigned based on many rules such as: unmatched word in both chunks is a number,either chunk has a token of LOCATION or DATE-TIMEtype. Ifpair chunks share one or more noun and Mikolov based similarity is >= 0.4 assign 3 score otherwise 2 score. The last one is if C-6 is not satisfied, score determined based on Milkov similarity [5]. REL relation type assigned if both chunks not share noun but share at least one content word. Scores are assigned as perMilkov similarity. NeRoSim limited alignment one chunk withanother one only, but if a chunk attempt to align with the one already aligned and has strong similarity of Milkov similarity, it assign ALIC relation with score of 0 [5]. Similarly, Venseseval system [17] built a system that is an adaptation of a pre-existing (VENSES) system, first makes analysis ofsemantic fora text including its structure and it looks for chunk linking information using knowledge resources.The Venseseval system takes pair of sentence, then select first chunk in first sentence and recursively attempt to match to every chunks in second sentence. For each chunk pair start matching procedures check from EQUI/OPPO then SPE1/SPE2 then SIMI/REL else assign NOALI label to chunk of sentence one and move to next. Similarly it repeats up to end of chunks in sentence one. Finally, the algorithm checksall NOALI marked chunks for possible multiple align matches with all already matched chunks except chunks labeled EQUI[17]. Moreover, thus rule based algorithm is constructed from too much different rules for similarity score assign. Chunk matching use different resources at different levels. To determine EQUI and SIMI relation WordNet in the same synset,and path one level similarityperformed. VerbOcean3and thesaurus used to determine REL relations [17].

III. MACHINE LEARNING

This section discusses about methods of chunk alignment, scoring and extracting their interpretation based on ML approach. Usually this approach more focus on a syntactic form of a chunks, for examplecount of POS and/or the count of words in a chunk pair [13]. In order to align chunks monolingual word aligner supervised ML techniques as Sultan [18] was chosen by two teams named UBC and ExB [11], [8]. Likewise, SVCSTS team [13] extends the technique but, IISCNL team [19] proposed a novel algorithm named iMATCH for alignment which handles many-to-many chunkalignment, based on Integer Linear Programming. Unsupervised ML is not used by many authors as supervised one. Unsupervised ML extract a defined score from dataset and use it along with other features to train a model [8]. STS score is computed using ν-SVR with default SVR parameter settings via LibSVM's. Fails to differentiate SIMI and REL type [8]. On the other hand, supervised ML uses many features like length, counts of parts of speech, order of words ineach chunk in pair to assign a type.Count of nouns, verbs, adjectives and prepositions in both chunk taken into account. The path similarity between words of pair chunks. Unigram as well as bigram overlaps between chunk pairs considered to predict alignment type [13]. The UBC team built a cube with information from several sources including Random Walks over Wikipedia and WordNet (depth related features), string similarity (Jaccard overlap related features), numbers (segment length related features), negation and antonym. Support Vector Machine (SVM) implementation using randomly shuffled 5-fold cross validation used to induce the model [11]. Similarity, supervised multiclass classification based on Random Forrest Classifier assign type and score for aligned chunks.Chunk Length Difference, Common Word Count, Has Number, Is Negation, Edit Distance Score, PPDB Similarity, cosine of W2V and Bigram Similarity. Wordnet based feature like, Path Similarity, IsHyponym, Synonym and Antonym Count [19]. FBK-HLT-NLP group constructs expandable and scalable pipeline framework, in which each component produces diverse features autonomously and at the conclusion, all highlights are solidified by a ML tools, which learns a relapse demonstrate for foreseeing the likeness scores from given sentence-pairs. The framework built combining diverse linguistic features in a classification show for foreseeing chunk-to-chunk arrangement, connection type, and STS score. The framework adopted string likeness, character/word n-grams, and pairwise similitude in UKP; in any case, on best of that the creators include other recognized features, like information of syntactic structure, semantic word similarity, and alignment a total of 245 features [9].

Several WordNet based features evaluates the type of relation between chunks by considering all the lemmas within the two chunks and checking whether a lemma in chunk1 may be an antonym, synonym, hyponym, meronym, hypernym, or holonym of a lemma in chunk2 [9].

A distributional representation of the chunk adds up to 200 features for chunk match, to begin with calculating word embedding and after that combining the vectors of the words within the chunk Mikolov word2vec with 100 dimensions [9]. WEKA was utilized for learning a relapse show to foresee the likeness scores a ML toolkit. Exploit the syntactic information by the mean of three particular toolkits: Syntactic Tree Kernel, Distributed Tree Kernel, and Syntactic Generalization. Then combines the yield of the three classifiers organized in a pipeline. For each adjusted chunk match, it includes the type and the STS score [9].

IV. DEEP LEARNING

This section discusses about deep learningand neural network based chunk aligning, scoring and labeling their relation. Chronological order of alignment prediction was strictly considered as in Sultan [18]. In [12] system the alignment began with token to token matrix performed on weighted sum of lowercased, stemmed or lemmatized token overlap, and cosine similarity between Mikolov's vectors. Once the token-token network is built, the alignment component makes utilize of fragment locales to gather every token. By carrying out this operation over all portions of the combine the module gets the chunk-chunk matrix. Once the matrix has been calculated, the final step is finding the sections (x, y) that maximize the association weights [12].

On different aspect neural network based approach (normal arrow) left and right segments are processed through a recurrent ANN generating as output a d-dimensional vector for every enter segment stated in [12]. Features computed out of those vectors are then fed to each a regressor and a classifier that produce the similarity rating and the relation label. In the backward propagation, weights are adjusted in the recurrent ANN combining the gradients that propagate from the models. As a two-layer architecture, a classifier and regressor work on the top of a recurrent ANN [12]. While the models at the top layer are trained to provide scores and labels, the underlying recurrent net attempts to capture the semantic representation of entering segments and feed it upwards.

Both models on the top layer are simultaneously trained in a supervised manner, and the delta error messages computed on them are used to train the bottom layer net. That is, to train the ANN's weights, the gradient propagating from both models on the upper layer was being used. The model works in the following way: one at a time, the ANN from the bottom layer processes fragment words and continues the same technique until no more words are left. At any specific time, the net updates its internal memory state, so that the semantic recognition of the segment continues to be captured[12].

When the two segments have been processed, the net results are d-dimensional vectors of segment representation. In the upper layer, these vectors are used to compute features for models. The two ANN models (RNN and LSTM) are coded according to the [25] equations. The concatenation of distance and angle gives 2 * d-dimensional vector. This resulting vector is used as the input in top layer models. Feed forward neural networks are used with relation to the upper layer models [12].

As a means of making a system more interpretable, the splitting of sentence level scores through subsequence alignments has been suggested. Predicting an agreement between chunks of sentence x and sentence y is the issue of interpretable STS. In sentence x, not all chunks are matched with a chunk in sentence y (and vise-versa). It was pointed out that a novel pointer network based alignment model was introduced in a recent study to align constituent chunks that are represented using BERT. Chunk representation is obtained from BERT[22] based on a chunked sentence, and by concatenating context - dependent embedding between the first and last word of a chunk. Word matrices are of the same dimension and the embedding of the project chunk into a lower vector dimension. The PN 'points' from chunks in x to chunks in y, thus. The system alignments are bidirectional to penalize misalignments in pairs of sentences. Guiding neural networks with integrated external sources has been shown to increase prediction accuracy by combining effective data-driven learning.

Two intuitive principles were employed for the chunk alignment phase.First relation rule is obtained from ConceptNet (i.e. Antonym, Synonym, IsA, RelatedTo, SimilarTo, DistinctFrom or FormOf) and the second rule is syntactical similarity (Jaccard similarity between POS tags of ancestor/children nodes two words) of two sentences as dependency parse trees [20].

V. HYDRIDE

Let, look at rule-based approach blend with ML approach. The VRep methods merges the two approaches and extracts for each chunk pair a total of 72 syntactic and semantic factors. VRep combines the ideas of NeRoSim's and SVCSTS's. NeRoSim is a rule based on the semantic link between chunk pairs, examining that two chunks contain antonyms, synonyms, etc. The methods of SVCSTS pay attention to the syntactic forms, while counting parts of speech and the number of words addressed in the ML approach in a chunk pair. Both these systems identify a chunk by using attributes derived from the chunk pair itself [21]. Gold standard chunk pairs of task 2 test data from the 2015 SemEval were used to learn their classifier, which generates a classification decision list. The classifier usedmany features and a series of rules. Classifiers were trained with chunk pairs from every data set (student answers, headlines, and images), both individually and combined [21].

UWB team [23] won in SemEval-2016 competition in the Gold standard chunk scenario. UWB is paired with a wide range of different models and aspects of similarity. In the method, ML and rule-based approaches to the task are explored [23]. More emphasis on ML and experimenting with a broad range of ML algorithms and also with many kinds of features based on rules. Four categories of features (lexical, syntactic, semantic, and external) employed. Lexical features consist word lemma overlap, word base structure overlap, difference in chunk length, difference in word sentence positions.POS tagging and parsing are performed with Stanford CoreNLP [24] for syntactic features. If one chunk is associated with several chunks in the other sentence, these chunks should be combined into one chunk after processing. To evaluate the similarity of chunks of a sentence the core of the method is to use distributional semantics. The authors of UWB method used the chunk similarity as a feature in unsupervised ML approach. The authors employ classification frameworks for chunk alignment, score and type classification by Voted perceptron (Weka), Maximum entropy (Brainy), and Support vector machines (Brainy) respectively. Little options used in rule-based approach which contributes to achieve best results for calculating chunk similarity with Word2Vec and the modified lexical semantic vectors. Type classifications classify all matched pairs of chunks as per predefined set of types [23]. In another work, for Indonesia a researchers adopt two best technique of SemEval-2016 named UWB which uses word embedding [23] and VRep that utilized WordNet to represent word semantic [21]. The adaptation of UWB and VRep is performed by changing English resources (such as word embedding and WordNet) in Indonesia [2].

VI. APPLICATION OF INTERPRETABLE STS

It is very important when learners engage with the application of an intelligent tutoring system through natural language explanation[1].In NLP, the scientific discovery of semantic similarity in text is very relevant and widely studied, with various tasks such as entailment, semantic similarity, etc.

Specificallyto clarify an advantage of interpretableSTS application, judgments to humans is required, it was pointed out that, in [27] performed two user studies on English text. In this work the authors first developed a verbal expression algorithm that returns the text verbalizing their commonalities or differences between the two sentences, given the pair of sentences, their similarity score, as well as the dictated and scored alignment between pairs of chunks. After that, differentiated the activities of the users without and with the Interpretable STS verbalizations. Finally, the results shows that Interpretable STS explanations are effective in both studies [27].

In any application, Interpretable STS can be applicable if showing additional explanation of similarity gives motivation and answer why x is similar with y. Even in recommendation system explanation of why the system recommend something or some product is valuable.

VII. CONCLUSION AND FUTURE WORK

To conclude let, look at state-of-the-art on allevaluation methods such asalignment, score, type prediction and overall results. BERT based chunk alignment is the best one by alignment of 97.73% on headlines and 96.32% on images dataset regardless of any further description about its chunking standard. The next table shows the best result of three datasets in two known chunk standard which presented on SemEval-2016. To show the overall results across datasets for score& type twogroups UWB and DTSim won for the gold chunks scenario, and another two groups FBK-HLT-NLP and DTSim wonfor the system chunks scenario. Moreover, DTSim obtained the best overall results but,from Answer-Students dataset the archived result was not good. The following table contains state-of-the-art on three dataset (Headline, Image, Answer-Student) with two chunks (GoldChunk and SysChunk).

Deep neural network BERT based alignment accuracy is more than 96% however, this method is not trained for score calculation. Score calculation result as state-of-the-art is less than 85% it shows as it needs working onit. Similarly type prediction is still less than 75% also the overall interpretableSTS result isless than 75%.Toimprove overall performance of the interpretableSTS deep learning approach is promising.

Table-I:state-of-the-art on align, score, and type prediction

Dataset name	Alignment	Score	Type	Score &Type
Images Syschunks	IISCNLP	FBK-HLT-NLP	DTSim	DTSim
	84.6%	78.6%	62.8%	61%
Images	UWB	UWB	UWB	UWB
Goldchunks	89.4%	84.1%	68.7%	67.1%
Headlines	DTSim	DTSim	DTSim	DTSim
Syschunks	83.8%	76%	56.1%	54.7 %
Headlines	IISCNLP	UWB	Inspire	Inspire
Goldchunks	91.4%	83.8%	70.3%	69.6%
Answer-Students	DTSim	FBK-HLT-NLP	FB-HLT-NLP	FBK-HLT-NLP
Syschunks	81.8%	75.9%	56.1%	55.5%
Answer-Students	VRep	IISCNLP	IISCNLP	IISCNLP
Goldchunks	87.9%	82.6%	65.1%	63.9%

REFERENCES

1. G. Majumder, P. Pakray, D. Avendaño, "Interpretable Semantic Textual Similarity Using Lexical and Cosine Similarity," Social Transformation – Digital Way. CSI 2018. Communications in Computer and Information Science, vol 836. Springer, Singapore. https://doi.org/10.1007/978-981-13-1343-1_59
2. R. C. Rajagukguk, M. L. Khodra, "Interpretable Semantic Textual Similarity for Indonesian Sentence," 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), 2018, pp. 147-152
3. Abney, S. 1996. Tagging and Partial Parsing, Corpus-Based Methods in Language and Speech. Kluwer Academic Publishers, Dordrecht
4. Nay Lin, et al: "Developing a Chunk-based Grammar Checker for Translated English Sentences," 25th Pacific Asia Conference on Language, Information and Computation, pp. 245–254.
5. R. Banjade, et al. "NeRoSim: A System for Measuring and Interpreting Semantic Textual Similarity," Proc. 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 164–171.
6. Chang, C.C., Lin, C.J, "LIBSVM: A library for support vector machines," ACM Trans. On Intelligent Systems and Technology2, 27:1–27:27(2011)
7. Ana O, et al. ASAP-II: From the Alignment of Phrases to Text Similarity. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 184–189.
8. C. Hanig, R. Remus, X. D. LaPuente, "ExB Themis: Extensive Feature Extraction from Word Alignments for Semantic Textual Similarity," Proc. 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 264–268.
9. N. P. An, S. Magnolini, O. Popescu, "FBK-HLT: A New Framework for Semantic Textual Similarity," Proc. 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 102–106.
10. M. Kazmi, P. Schüller, "Inspire at SemEval-2016 Task 2: Interpretable Semantic Textual Similarity Alignment based on Answer Set Programming," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 693–699.
11. Eneko Agirre, et al. "UBC: Cubes for English Semantic Textual Similarity and Supervised Approaches for Interpretable STS," Proc. 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 178–183.
12. I. Lopez-Gazpio, E. Agirre and M. Maritxalar, "iUBC at SemEval-2016 Task 2: RNNs and LSTMs for interpretable STS," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 681–686
13. S. Karumuri, V. K. Reddy and S. C. Raj, "UMDuluth-BlueTeam: SVCSTS - A Multilingual and Chunk Level Semantic Similarity System," Proc. 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 4-5, 2015, pp. 107–110.
14. R. Banjade, N. Maharjan, N. B. Niraula and V. Rus, "DTSim at SemEval-2016 Task 2: Interpreting Similarity of Texts Based on Automated Chunking, Chunk Alignment and Semantic Relation Prediction," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 726–730.
15. Tomas Mikolov, et al. "Efficient Estimation of Word Representations in Vector Space," Proc. Workshop at ICLR, 2013
16. T. P. Ping, K. Verspoor and T. Miller, "Rev at SemEval-2016 Task 2: Aligning Chunks by Lexical, Part of Speech and Semantic Equivalence," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 777–782.
17. R. Delmonte, "VENSESEVAL at Semeval-2016 Task 2: iSTS - with a full-fledged rule based approach," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 713–719.
18. Md. A. Sultan, S. Bethard, and T. Sumner, "Dls©cu: Sentence similarity from word alignment," Proc. 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 2014, pp. 241–246.
19. L. S. Tekumalla and Sharmistha, "IISCNLP at SemEval-2016 Task 2: Interpretable STS with ILP based Multiple Chunk Aligner," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 790–795.
20. S. Maji , R. Kumar , M. Bansal , K. Roy and P. Goyal, "Logic Constrained Pointer Networks for Interpretable Textual Similarity," Proc. Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20).
21. S. Henry and A. Sands, "VRep at SemEval-2016 Task 1 and Task 2: A System for Interpretable Semantic Similarity," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 589–595.
22. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," In NAACL-HLT, 2019, pp. 4171–4186.
23. M. Konopik, O. Prazak, D. Steinberger and T. Brychcin, "UWB at SemEval-2016 Task 2: Interpretable Semantic Textual Similarity with Distributional Semantics for Chunks," Proc. SemEval-2016, San Diego, California, June 16-17, 2016, pp. 720–725.
24. Christopher D. Manning, et al. "The Stanford CoreNLP natural language processing toolkit," Proc. ACL: System Demonstrations, 2014
25. Kai Sheng, et al. "Improved semantic representations from tree-structured long short-term memory networks," 2015.
26. Dan Stefanescu, et al. A sentence similarity method based on chunking and information content. In Computational Linguistics and Intelligent Text Processing, pages 442–453. Springer. 2014b.
27. I. Lopez-Gazpio, et al. Why do you say they are similar? Interpretable Semantic Textual Similarity

AUTHOR PROFILE

Abdo Ababor Abafogi: Received his BSc and MScin Information Technology, from Jimma University, Ethiopia in 2012 and 2017 respectively. His research interest areas are Artificial Intelligence, Data Science, Deep Learning, Machine Learning and Natural Language Processing