An essay on paraphrase identification using machine learning and deep learning approaches

The ability to paraphrase can be attributed as evidence that demonstrates an understanding of any text. It can be an important tool to convey complex ideas in a different manner to make them easier to understand [1,2]. The ability to identify a paraphrase is just as important as it can help to measure semantic equivalence between texts for NLP tasks such as information extraction, text summarization, machine translation, and question answering [3,4]. Paraphrases can take many forms which can be a major problem with their identification since models accounting for all those forms would be required to resolve the identification. For example, consider the sentence "Symptoms of influenza include fever and nasal congestion." A paraphrase for this sentence would be "A stuffy nose and elevated temperature are signs you may have the flu." [5]. It can be observed from the above example that the paraphrased sentence has completely different words than the original but they have the same meaning. This is an example of a higher-degree paraphrased sentence that would be difficult to identify using models that use word-matching techniques. The model also has to account for the semantic relationship between texts, for example, by keeping the track of synonyms, since it is preserved in the paraphrased sentence [6]. Consider another example, "I need to throw away tons of stuff" and the paraphrase, "I need to throw away a lot of things". While some words are changed, a word matching model could work for documents like these to identify the paraphrase. This essay focuses on various approaches for paraphrase identification that uses different text similarity measures with machine learning, and deep learning models. The dataset considered here is the Microsoft Research Paraphrase Corpus (MSRP) which is evaluated in all of the approaches that are the focus of this essay. It contains 5,801 sentence pairs extracted from news articles. Each sentence pair contains a human-annotated label that denotes if the pair has paraphrase equivalence or not [7].

There are multiple text similarity measures such as lexical similarity, term-frequency-based similarity, word order similarity, and semantic similarity to name a few [8]. The lexical similarity measure can be a good candidate to Identify paraphrases between a sentence pair as by definition, it means the measure of overlapping of word sets [9]. Since paraphrases can have overlapping words with their original counterpart, various lexical matching features could be used to find their similarities and classify them based on paraphrase equivalence. Zhang and Patrick [3] show a similar approach with a supervised learning method using a decision tree to classify if sentence pairs from the MSRP dataset have paraphrase equivalence or not. First, they transform the sentence pairs into a canonical form using a limited set of rules to regulate their format. Specifically, the transformation is replacing the numerical entities with a generic tag, changing passive voice to active voice as it is more consistent with the broader Subject-Verb-Object structure in non-formal English texts, and replacing future tense keywords like "plan to" and "expected to" with "will". Four lexical matching features namely, "Longest common substring" which is the measure of the longest strings appearing in both the sentences, "Longest common subsequence" which measures the longest common sequence appearing in both the sentences, "Edit distance" which measures the number of add, delete or replace operations required to convert the original sentence to the paraphrased sentence, and "Modified N-gram precision" that measures the relationship between the original and paraphrased sentence using minimum and maximum n-gram count. These features combined with the canonicalization of the sentence pairs represent different ways word sets can overlap. They are used to train a decision tree to predict if a sentence pair shows paraphrase equivalence. The results were obtained for an individual as well as combined canonical rules. Changing the passive to active voice rule achieved the highest scores with an accuracy of 71.9%, precision of 74.3%, recall of 88.2%, and F1 score of 80.7%. The results could be attributed to the limited set of canonicalization rules and the model only accounting for the lexical similarity measures i.e. quantitative word features such as length without considering the lexical-semantic similarity measure i.e. use of synonyms and word order between word sets. For example, consider the sentence “Will slapped Chris at the Oscars.” and another sentence “Chris slapped Will at the Oscars.” if lexical similarity is the only concern, both the sentences are similar as the length of the longest strings and the longest sequence is similar. Also, a minimum amount of editing operations is required to convert one sentence to the other. However, both the sentences have completely different meanings.

To account for semantic similarity between texts, methods like latent semantic analysis can be used. LSA is used to extract contextually similar concepts that can identify a topic of interest. It uses statistical computations to analyze the co-occurrence of words within a document with an assumption that the words that frequently appear together in a similar context, have similar meanings [10]. The central process in LSA is matrix decomposition. Initially, the process starts with a document-term matrix that has the document list as rows and all the tokens/terms as columns. Matrix decomposition is required as the document-term matrix is very sparse and this process reduces it into multiple orthogonal matrices that can be used to approximate the original sparse matrix. The co-occurrence matrix values are used in matrix factorization to obtain semantically similar concepts that can approximately identify a document. Ji and Eisenstein [11] showed how latent vectors i.e. the output of LSA can be used as input to a supervised learning algorithm. For the paraphrase identification task, the matrix *MxN* (where M contains the sentences and N contains the features which can be n-grams) is decomposed to form a latent representation of semantically similar sentences by using non-negative matrix factorization. Since the matrix *MxN* is simply the count of each term *N* appearing in document *M*, they have proposed a ranking method “TF-KLD” that reweighs the counts of terms *N* according to their importance by combining term frequency and Kullback-Leiber divergence before the factorization process. Rare or discriminative words are given more importance than frequently occurring words. The final matrix is a semantic latent vector that is reweighed according to the term importance and it is the input to an SVM model. Since this task is a supervised learning task, the authors used the advantage to add additional features from Wan et al. [12] that captured more detailed lexical similarities between sentences by calculating unigram and bigram overlap. The authors evaluated the model in two sets, using only unigrams as the feature *N*, and using unigrams, bigrams, and unlabelled dependency pairs obtained from MaltParser [13] as the feature *N*. A dependency pair as the name suggests represents dependencies between two words in a text. For example, consider two words “crying baby”, the word “crying” is affecting the word “baby” hence, the word “crying” is the dependent word [14]. The best results were obtained by using unigrams, bigrams, and dependency pairs with an accuracy of 80.41% and an F1 score of 85.96%. Using LSA helped to identify the semantic features, and combined with the lexical similarities from Wan et al., the scores improved a lot over [3] that just used lexical similarities.

With the advancement of deep learning techniques, word embeddings are heavily being researched in NLP. Word embeddings are fixed dimensional vector representations of words. The idea behind word embeddings was to model distributional semantics meaning, that similar word vectors should occupy similar vector spaces. Hence, various similarity measures such as cosine similarity, Euclidean distance, and other arithmetic operations can be used to manipulate the vectors [15]. Word2Vec initiated the idea of pre-training models on large datasets that can be used for transfer learning by training neural networks to predict the most likely word given the context [16]. GloVe followed a year later by taking into account the word co-occurrence over the whole corpus and finding the probability of two words appearing together [17]. Sequence models such as RNNs, LSTMs, and GRUs are primarily used in NLP as they have the ability to keep track of any sequence and record the context of each state in a sequence as hidden states that can be passed into the next state in the time step. Hence, they can account for semantic and syntactic similarities [18]. Shen et al. [19] use a Bi-directional LSTM-gated relevance network with Word2Vec embeddings pre-trained on 100 billion tokens from Google News data for paraphrase identification. They chose LSTM as it solved the problem of vanishing gradient in RNNs due to which the context present in the hidden states of the earlier time steps is lost. Both the original and the modified sentences are provided as input to the bi-directional LSTM. The gated relevance network computes the relevance score between the intermediate positional encodings i.e. “interaction” of the two sentences. The final output of the GRN is a relevance matrix that contains values for each time step. This matrix is passed through a max-pooling layer that extracts the most relevant interaction. The output of the pooling layer is reshaped and fed to a multi-layer perceptron network to classify if the sentence pair has paraphrase equivalence. The authors used the following hyper-parameters: Batch Size = 64, Word embedding dimensions = 300, pooling stride = (3,3), tensor slices = 2, learning rate = 0.01. This model produced the best results with an accuracy of 80.92% and an F1 score of 86.5%. The approach corroborates the results as the pre-trained Word2Vec word embeddings used for transfer-learning were trained on a large corpus which increases the probability that the model has data for the vocabulary of the test set. Also, since LSTMs account for both syntax and semantics, it further supports the findings from [11] and the importance of accounting for semantic similarity.

To summarize, three approaches were discussed for paraphrase identification on the MSRP dataset. Zhang and Patrick [3] only used lexical similarity measures while the model by Ji and Eisenstein [11] accounts for both lexical and semantic similarity, and the model by Shen et al. [19] accounts for syntax and semantics. [11] and [19] showed comparable performance and both outperformed [3]. It is worth noting that the score comparison between [11] and [19] is on-par despite the former not using any large pre-trained word embeddings. This highlights the robustness of the model proposed by Ji and Eisenstein as it is performing similarly to [19] with no extra information. Does it mean the problem of paraphrase identification is solved? Weeds et al. pointed out that the MSRP dataset is not a good source to study paraphrase identification because of high word-set overlapping [20]. The same was acknowledged by Dolan and Brockett from Microsoft research [7]. This might be the reason for [3] to perform reasonably well by just considering lexical similarity. In the real world, however, paraphrases can contain words that do not share the same semantic meaning as the original sentence but the overall meaning of the sentence is preserved. The reason this dataset was chosen as a benchmark is that it is the most studied dataset since it has been around for a decade [21]. To conclude, the LSA model [11] has shown a lot of potential and it is the best approach in this study from my perspective for the reasons stated above. For future work, it would be really good to research solutions like the LSA model that can provide similar results as large pre-trained models. Maybe trying to combine the semantic latent vector from [11] and using it as input in the LSTM model from [19] might help resolve lexical similarity, semantics, and syntax together. It will also provide two representations of semanticity that can be tested for co-relation. This research direction will provide the opportunity for robust innovative solutions in the domain of artificial “understanding”.

References:

[1] G. A. Miller, WordNet: A lexical database for English, Commun. ACM 38(11) (1995) 39–41.

[2] Burnell, Carol, Jaime Wood, Monique Babin, Susan Pesznecker, and Nicole Rosevear. 2019. “Paraphrasing.” Pressbooks.pub. Pressbooks. 2019. https://openoregon.pressbooks.pub/wrd/chapter/paraphrasing/

[3] Zhang, Yitao, and Jon Patrick. ‘Paraphrase Identification by Text Canonicalization’. Proceedings of the Australasian Language Technology Workshop 2005, 160–66. Sydney, Australia, 2005. https://aclanthology.org/U05-1023.

[4] Rinaldi, Fabio, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. "Exploiting Paraphrases in a Question Answering System." *Proceedings of the Second International Workshop on Paraphrasing* -. https://doi.org/10.3115/1118984.1118988.

[5] YourDictionary. 2018. "Examples of Paraphrasing." YourDictionary. November 9, 2018. https://examples.yourdictionary.com/examples-of-paraphrasing.html.

[6] Kong, Leilei, Zhongyuan Han, Yong Han, and Haoliang Qi. 2020. "A Deep Paraphrase Identification Model Interacting Semantics with Syntax." *Complexity* 2020 (October): e9757032. https://doi.org/10.1155/2020/9757032.

[7] Dolan, William, and Chris Brockett. n.d. "Automatically Constructing a Corpus of Sentential Paraphrases." https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/I05-50025B15D.pdf.

[8] Achananuparp, Palakorn, Xiaohua Hu, and Xiajiong Shen. n.d. "The Evaluation of Sentence Similarity Measures." *Data Warehousing and Knowledge Discovery*, 305–16. https://doi.org/10.1007/978-3-540-85836-2_29.

[9] Majumdar, Dattatreya. 'Text Analysis and Distant Reading using R'. 2022.04.03. Brisbane, 2022.

[10] Landauer, Thomas K, Peter W. Foltz, and Darrell Laham. 1998. "An Introduction to Latent Semantic Analysis." *Discourse Processes* 25 (2-3): 259–84. https://doi.org/10.1080/01638539809545028.

[11] Ji, Y. & Eisenstein, J.. (2013). Discriminative improvements to distributional sentence similarity. EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 891-896.

[12] Wan, Stephen, Mark Dras, R. Dale and Cécile Paris. "Using Dependency-Based Features to Take the 'Para-farce' out of Paraphrase." *ALTA* (2006).

[13] NIVRE, JOAKIM, JOHAN HALL, JENS NILSSON, ATANAS CHANEV, GÜLŞEN ERYİGİT, SANDRA KÜBLER, SVETOSLAV MARINOV, and ERWIN MARSI. 2007. "MaltParser: A Language-Independent System for Data-Driven Dependency Parsing." *Natural Language Engineering* 13 (2): 95–135. https://doi.org/10.1017/s1351324906004505.

[14] Koo, Terry, Xavier Carreras, και Michael Collins. 'Simple Semi-supervised Dependency Parsing'. Στο Proceedings of ACL-08: HLT, 595–603. Columbus, Ohio: Association for Computational Linguistics, 2008. https://aclanthology.org/P08-1068.

[15] Almeida, Felipe, and Geraldo Xexéo. n.d. "Word Embeddings: A Survey." https://arxiv.org/pdf/1901.09069.pdf.

[16] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." ArXiv.org. 2013. https://arxiv.org/abs/1301.3781.

[17] Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "GloVe: Global Vectors for Word Representation." https://nlp.stanford.edu/pubs/glove.pdf.

[18] Chung, Junyoung, Caglar Gulcehre, and Kyunghyun Cho. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." https://arxiv.org/pdf/1412.3555.pdf.

[19] Shen, Yatian, Jifan Chen, and Xuanjing Huang. 2016. "Bidirectional Long Short-Term Memory with Gated Relevance Network for Paraphrase Identification." *Natural Language Understanding and Intelligent Applications*, 39–50. https://doi.org/10.1007/978-3-319-50496-4_4.

[20] Weeds, Julie, David Weir, και Bill Keller. 'The Distributional Similarity of Sub-Parses'. Στο Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, 7–12. Ann Arbor, Michigan: Association for Computational Linguistics, 2005. https://aclanthology.org/W05-1202.

[21] Rus, Vasile, Rajendra Banjade, and Mihai Lintean. 'On Paraphrase Identification Corpora'. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2422–29. Reykjavik, Iceland: European Language Resources Association (ELRA), 2014. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1000_Paper.pdf.