Can Machines Read Jmulbed Senetcnes?

Runzhe Yang
Princeton University
runzhey@cs.princeton.edu

Zhongqiao Gao
Princeton University
zg2@cs.princeton.edu

Abstract

This research is driven by our curiosity about the limits to which the current technology for natural language processing could be pushed: if humans are capable of reading jumbled sentences, can machines do? To answer this question, we evaluate the ability of state-of-the-art models to recognize and comprehend word- and character-level jumbled sentences, analyze factors that may influence machine's performance, and devise potential cures for enhancing models' robustness. The quality of sentences embeddings is investigated on three classic downstream tasks mimicking human cognitive abilities, including sentiment classification, information retrieval, and semantic similarity, to reflects machines' capability in reading jumbled sentences. We discover induced biases in embeddings for jumbling sentences which impairs machines performance. The removal of these induced biases significantly improves machines' robustness of reading character-level jumbled sentences on all three tasks.

1 Introduction

Recent decades have witnessed much convenience in our daily life provided by the advances of natural language processing (NLP) techniques, from instant online translators (Green et al., 2013) to electronic health records (Jacobson and Dalianis, 2016), and personal speech assistants (Chang et al., 2017). Mainly due to the emergence of deep learning and neural network based algorithms, researchers have claimed that machines achieved human-level performance or even beyond in many NLP application domains, such as speech recognition (Xiong et al., 2016), machine translation (Wu et al., 2016; Hassan et al., 2018; Klein et al., 2017), and question answering (Devlin et al., 2018; Socher et al., 2018). Do machines really rival humans in comprehending language? How do computers and humans differ in ways of processing language? Is there any key element which is very important for human language understanding but still missing in state-of-the-art NLP techniques? In this research project, we are aiming at answering broad questions above by investigating a specific cognitive task — can machines read jumbled words and sentences?

Many studies on psychology and cognitive science (McCusker et al., 1981; Mayall et al., 1997) suggest that humans are capable of recognizing and comprehending scrambled words and sentences, though many of us do not realize it.

> For emaxlpe, it deson't mttaer in waht oredr the ltteers in a wrod aepaprm, the olny iprmoatnt tihng is taht the frist and lsat ltteer are in the rghit pcale. The rset can be a toatl mses and you can sitll raed it wouthit pobelrm.

As we may find in the above paragraph[1], when letters in words are in disorder, we can still read it without much difficulty. Actually, this cognitive phenomenon is common not only among English speakers. For instance, in Chinese, similar cognitive phenomenon also exists in word-level perturbation[2].

If humans have this remarkable language ability as illustrated, to overcome the misspelling and understand the jumbled nonsense, are our state-of-the-art NLP systems, which hit human parity in many tasks, also able to deal with garbled words

[1] https://www.livescience.com/18392-reading-jumbled-words.html
[2] https://www.guokr.com/blog/443743/

and sentences? Since machines do not rigorously mimic our language processing, what components, in common, or unique to machines or humans, are essential to this ability?

Research in this direction has more practical implications besides the satisfaction of curiosity and the comparison between human and machine cognition. When deploying an NLP system online in a real application scenario, it is not always the case that all words in the feeding target text or queries are correctly spelled or in the right order. Therefore, pre-processing of the input text via typo correction and normalization is usually required (Sun et al., 2014). However, a general mechanism for machines to generalize to directly deal with raw text with out-of-vocabulary (OOV) words and disordered sentences is conceivably preferred, since the pre-process cannot guarantee a perfect input, and is hard to be fit for the online setting. In this research, we harbor an ambitious hope to pave a road to the understanding and design of this general mechanism.

In this research, we are interested in finding: (1) At which level of the jumbled text, can the most advanced neural NLP systems read and understand? (2) What factors may influence current neural NLP systems for achieving good performances when reading jumbled sentences? (3) How can we possibly improve the neural NLP systems to have more robustness of processing the real-life natural language with rich diversity?

We design experiments to answer this three questions by investigating the state-of-the-art BERT model (Devlin et al., 2018) for sentence embedding on different downstream tasks. The reason we particularly interested in sentence embedding is that word and sentence embeddings are the fundamental components of every neural NLP system (Pennington et al., 2014; Arora et al., 2017), and they best demonstrate machines' understanding of texts as their inner representations.

The sentences in test cases is jumbled in the letter- or word-level, at different levels of readability judged by a human. The quality of sentence sentences embeddings is investigated on three representative downstream tasks mimicking human cognitive abilities, including sentiment classification (Wang and Manning, 2012), information retrieval (Li and Roth, 2002), and semantic similarity (Cer et al., 2017), to reflects machines' capability in reading jumbled sentences.

We discover induced biases in embeddings for jumbling sentences which impairs machines performance. Our experiments shows the removal of these jumbling induced biases significantly improves machines' robustness of reading character-level jumbled sentences on all three tasks.

2 Background

2.1 Psycholinguistics

To understand why human can overcome the typo noise and scrambled words, psycholinguists developed several hypotheses. The first hypothesis is about the strong language priori (Griffiths et al., 2008). Since humans have strong priori on what a "typical" word or sentence should be, we can match quickly from error form to similar correct form and then understand them correctly.

Moreover, the parallel processing hypothesis (Townsend, 1990) and unit processing hypothesis (Davis, 2004) maybe another factor for us to consider. Psychological experiments (Rawlinson, 1976) showed that in most cases only the first and the last letters matter in word recognition. The experiment conducted by (McCusker et al., 1981) also showed that swapped middle characters in a word went unnoticeable by the human. (Mayall et al., 1997) suggested that the shape of word plays a significant role in word recognition instead of the actual order. Humans may use parallel processing to manage multiple stimuli at the same time of different quality. Those stimuli will be analyzed separately, compared to prior knowledge, and finally combines the information in different weights, to give brain with the most helpful information.

2.2 Adversarial Examples

Why do we care about the design of a robust model to deal with noisy text? Many language models assume that the input to the models is in a natural or well-behaved distribution. However, their assumption may not hold in security-sensitive setting (Biggio et al., 2012). It is quite dangerous to have a machine learning system used without handling the noise input (Goodfellow et al., 2014). Even small imperceptible perturbation on the input data, would cause huge prediction error (Szegedy et al., 2013) (Mei and Zhu, 2015). On the other side, those imperceptible perturbation errors can be easily made by a human without notice, yet may cause misclassification in the language model.

2.3 Related Work

Previous studies researches how the model handles noisy text on the applications of information retrieval and information extraction (Subramaniam et al., 2009). They defined different types of noisy text and then surveyed some methods to overcome this issue. However, it is not a detailed survey containing standard test data for different methods, the only limited field of methods are investigated and have no detailed analysis on advantages, disadvantages and how to improve based on current design. (Belinkov and Bisk, 2017) conducted a character-based CNN that handles well on machine translation tasks with black-box adversarial training. (Sakaguchi et al., 2017) designed a robust semi-character recurrent neural network model to handle misspelled input text and received great accuracy with jumbled text.

Our research overcome the above issues to provide a detailed investigation on a state-of-the-art sentence embedding model incorporating the advantage of language modeling, to figure out whether the model is capable of extract information correctly even with human typos, or what specific factors may influence the model's performance. We use the same test sets in the original paper, process the test set to make jumbled sentences, in order to identify the effects of jumbled words and sentences to the models, and how to improve the model's performance based on current design and insights from these observed effects.

3 Methods

We designed a bunch of experiments to explore whether machines, which use the state-of-the-art BERT model for sentence embedding, can read jumbled sentences. In order to compare machines performance on the original text and the jumbled text, we designed several jumbling algorithms, performing 'swap', 'omit', and 'add' as basic operations to jumble the text in the character level as well as in the word level. We chose three downstream benchmark tasks for evaluating machines' ability to "read" this sentence: Customer Review (CR) (Wang and Manning, 2012), Text REtrieval (TREC)(Li and Roth, 2002), and Semantic Text Similarity (STS) (Cer et al., 2017). CR focuses on sentiment analysis of customer products' reviews, giving outputs of positive or negative. TREC classifies the questions into different categories (e.g., "What are the twin cities?" should be classified as `LOC: City`). STS measures the semantic similarity between two sentences, evaluated as a number from zero to five, i.e., not similar to very similar. We employ a benchmark toolkit for universal sentence representation (Conneau and Kiela, 2018) to simplify the process of unifying data formats and training and test pipeline. After testing the model, we compared the performances on different tasks under the character-level and the word-level jumbling schemes of different degree of disorder. We also visualize the jumbled sentence embedding using t-stochastic neighbor embedding (t-SNE) (Hinton and Roweis, 2002) and principle component analysis (PCA) (Wold et al., 1987) to investigate whether the jumbled sentences embedding shift with a biased or just randomly distributed compared to the normal.

3.1 Pre-training of Deep Bidirectional Transformers (BERT)

We use the state-of-the-art sentence embedding model in our experiment, pre-training of deep bidirectional transformers (BERT) (Devlin et al., 2018). BERT learns a bidirectional word representations, which is jointly conditioned on both left and right context in all layers when pre-train on deep bidirectional representations. The model fine-tuned with one additional layer on a wide range of tasks. This state-of-the-art model improves the GLUE benchmark to 80.4% with 7.6% absolute improvement. In our experiment, we use a cased English model pre-trained by Google, which has 12 layers, 12 heads, and about 110M parameters. The size of embedding is 768 in all our experiments.

3.2 Jumbling Schemes

We designed several jumbling schemes, which performs `swap`, `omit`, and `add` (repeat) operations in character level and word level on the text. The description of these schemes are in Table 1, and some examples of resulting jumbled sen-

Mode	Function
`swap`	randomly swap two adjacent words with prob=x
`omit`	randomly delete words with prob=x
`add`	randomly repeat words with prob=x

Table 1: Character-level and word-level jumbling schemes in `swap`, `omit`, and `add` modes.

Operation	Example (word-level jumbling)	Example (character-level jumbling)
Swap	What considered the is costliest disaster insurance the industry has ever ? faced	What is consdiered the csotliset disasetr the inuransce idnustry has ever faedc ?
Omit	What considered the costliest disater the insurance industry faced ?	What is consiered the cosliest saster the isrance industrindustry has ever faced ?
Add	What is considered the costliest disaster disaster the insurance industry has ever faced ?	Whapt is considerehd thes costlciessteb disasterb the insurancet industrydu has everyu faced ?

Table 2: Examples of processed sentences under word-level and character-level jumbling schemes with prob=0.2. Original: *"What is considered the costliest disaster the insurance industry has ever faced ?"*

Dataset	Task Description	Example Text	Label
CR	Sentiment analysis of reviews	*We tried it out Christmas night and it worked great*	Positive
TREC	Question Answering	*What are the twin cities?*	LOC:city
STS	Measuring the semantic similarity	{*Liquid ammonia leak kills 15 in Shanghai*, *Liquid ammonia leak kills at least 15 in Shanghai*}	4.6

Table 3: Task descriptions for Custom Review dataset, Text REtrieval Conference, and Semantic Text Similarity with example input sentence and the corresponding ground truth label.

tences are shown in Table 2. In our experiment, we set the probabilities of jumbling schemes at $\{0.20, 0.35, 0.50, 0.65, 0.8\}$ to simulated different degree of confusion.

3.3 Downstream Tasks

To evaluate the ability of BERT model to read jumbled texts, we test BERT in three downstream tasks: Customer Review (CR) (Wang and Manning, 2012), Text REtrieval Conference (TREC) (Li and Roth, 2002), and Semantic Text Similarity (STS)(Cer et al., 2017). CR task is a binary classification task about sentiment analysis of customer products reviews, where the machine is required to predict positive or negative after reading a user review. We tried to see how much jumbled character or word will affect the model's ability to determine sentiment information. The CR dataset we use contains 2406 positive examples and 1367 negative examples. TREC task is an information retrieval or question answering task, where the machine aims to classify the user questions into six categories: `ABBR`, `DESC`, `ENTY`, `HUM`, `LOC`, `NUM`. It tests model's ability to classify semantic information of the jumbled text. We use 5500 labeled sentences for training and 500 sentences for testing. STS task measures the semantic similarity between two paired sentences in a scale from 0 to 5. We used this task to evaluate the machine's ability to capture similarities between sentence embeddings of jumbled texts. There are 5749 training sentences, 1500 validation sentences, and 1379 test sentences in the STS benchmark dataset. Examples of tasks are illustrated in Table 3. We use PyTorch to train logistic regression models for CR and TREC in the 10-fold fashion.

4 Experimental Results

4.1 Effects of Jumbling Levels and Degrees

Our first experimental goal is to investigate at which level of the jumbled sentence the machine can understand. The results of performance comparison among different jumbling schemes are shown in Table 4, where the jumbling probability is fixed as 0.2. As we expect, the machine got all the best results on three tasks when reading original text without jumbling. Note that the original text may have only natural error inside the text. The word-level jumbling will not impair the performance too much on classification tasks. There is only about 2% drop of the accuracy on CR task, and 1% drop of accuracy on TREC task for word-level `swap` and `omit`. The Pearson correlation coefficient in STS task with human similarity evaluation is about 0.53, which still indi-

Operation	CR	TREC	STS
Random (baseline)	61.86	22.0	-0.014
Swap(char)	71.55	67.8	0.246
Omit(char)	69.03	68.4	0.286
Add(char)	69.91	66.4	0.293
Swap(word)	82.46	90.4	0.538
Omit(word)	82.68	90.4	0.530
Add(word)	82.70	87.0	0.536
Original	84.64	91.0	0.604

Table 4: Thes test performance comparison between word-level and character-level jumbled sentence embeddings, original sentence embedding and random embedding, evaluated on three different downstream tasks with a jumbling probability 0.2. For tasks CR and TREC, the displayed numbers are accuracy. For STS task, the shown numbers are Pearson correlation coefficients, which are the higher the better.

cates relatively strong correlation. The unsupervised pre-trained feature in BERT helps the model to still able to locate useful semantic and syntactic information in word-level jumbled sentences. As for the character-level jumbled sentences, the machine's overall performance on comprehending these sentences in all three tasks is much worse than that on the word-level jumbled sentences. However, the machine still can "read" this sentences compared to the random embedding baseline. Note that character-level `omit` and `add` have slightly better result than `swap` in the STS task in terms of predicting sentences' similarity. One possible reason is that the model incorporates word-piece embeddings (Devlin et al., 2018), which divides words into sub-word units in order to handle rare words. Therefore, having less information to divide into sub-word performs better than swapped sub-words in similarity comparison task.

We further study how the machine's performance varies when the sentence is getting more jumbled for both word and character level. Figure 1 shows results as curves of the test performance as the jumbled probability increases from 0.2 to 0.8. All the test performance on the jumbled sentences are worse than the unjumbled sentences jumbled while better than the random embedding baseline, which indicates it is difficult for machine to extract useful semantic information from the jumbled sentences than original unjumbled sentence, and partial information may be

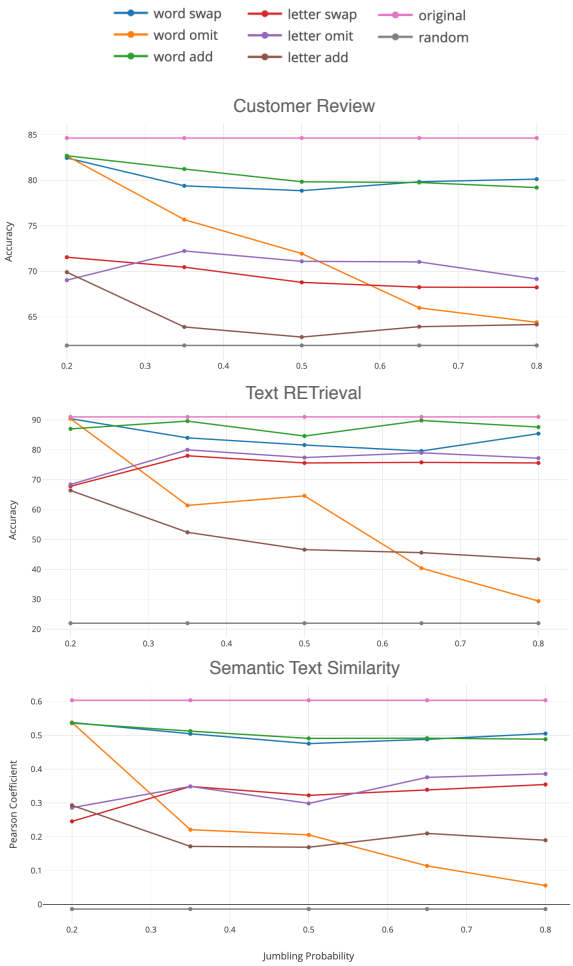

Figure 1: Curves of the test performance on different tasks as the probability of jumbling increases from 0.2 to 0.8. The test performance on character-level jumbled sentences are generally worse than that on word-level jumbled sentences for all three tasks. When sentences get more jumbled using word-level `omit`, the test performance on all three task drops rapidly.

lost, however, the machine is still able to extract some of the rest information. The test performance on character-level jumbled sentences are generally worse than that on word-level jumbled sentences for all three tasks, which is consistent with our previous result. When sentences get more jumbled using word-level `omit`, the test performance on all three task drops rapidly. This could be explained as the more word deleted from sentence, the more semantic information would be lost. The test performance only slightly changes as sentences get more jumbled using other jumbling schemes, which suggests the machine's ability to extract information from jumbled sentence

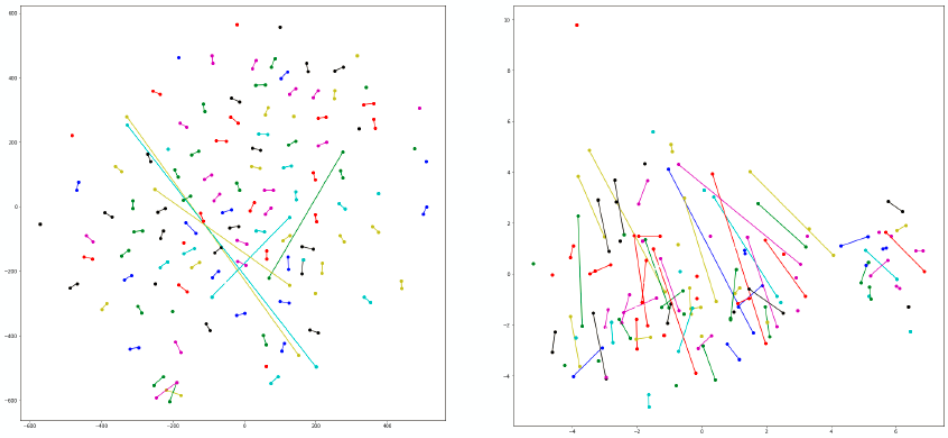

Figure 2: Word-level `add` jumbling operation on CR task in word level with t-SNE and PCA

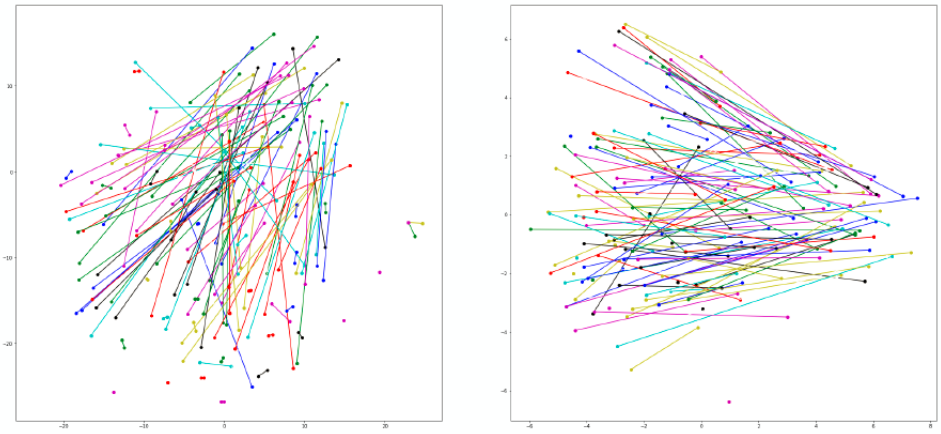

Figure 3: Character-level `add` kumbling operation on CR task in word level with t-SNE and PCA.

is very limited, and slightly losing more informations from the sentence will not affect the test performance too much since the main restriction of performance is the machine's capacity of capturing semantic information from jumbled sentence instead of competency or clearance of sentences.

4.2 Visualization of Jumbled Sentence Embedding with Dimension Reduction

In order to understand better about the influence of the word and character level jumbling, we then implement t-SNE(t-distributed Stochastic Neighbor Embedding) and PCA(principal component analysis) visualization of the sentence embeddings. t-SNE will preserve local geometry relationship between original and jumbled text sentence embeddings. PCA will preserve distances and angles. Figure 2 shows examples of visualizing jumbling operation `add` on the CR task in word level. Figure 3 shows visualizing `add` jumbling operation on the CR task in character level. From the t-SNE visualizations, we can see that character level jumbling is significantly more violent than word level jumbling. As we expect that jumbling characters will deviate more from the original text. For the PCA visualizations, we find obvious induced bias in both jumbling embedding. These obvious biases inspire us that if we deduct the reversed bias from the jumbled sentence embeddings, will the new sentence embeddings improves the accuracies of the jumbled text?

4.3 Effects of Induced Biases: A Simple Cure

To check our induced bias assumptions, we calculate the average difference between original text and jumbled text after convert them into lower dimensions using PCA. We then convert the bias back into the original dimension and then subtract the it from the jumbled sentence embeddings. The result of the induced biases for jum-

	CR	CR-bias	TREC	TREC-bias	STS	STS-bias
Original	84.93		90.8		0.6036	
Swap (w)	84.43	62.28	89.2	89.6	0.537	0.475
Omit (w)	82.62	82.33	89.2	89.6	0.523	0.325
Add (w)	81.74	82.44	91.2	91.4	0.523	0.514
Swap (c)	69.04	**81.86**	69.6	**86.8**	0.219	**0.463**
Omit (c)	68.74	**79.44**	70.8	**79.8**	0.223	**0.369**
Add (c)	70.52	**83.36**	67.4	**90.0**	0.206	**0.537**

Table 5: The test performance compares the accuracies between jumbled sentence embeddings and reversed bias sentence embeddings for both word-level and character-level jumbling. For tasks CR and TREC, the displayed numbers are accuracy. For STS task, the shown numbers are Pearson correlation coefficients, which are the higher the better.

bling CR, TREC, STSBenchmark task in both word and character level are shown in Table 5. There are significant improvement in the character level jumbling, as for word level Swap jumbling bias reverse, the accuracies improves from 69.04 to 81.86, Omit jumbling improves from 68.74 to 79.44 and Add jumbling improves from 70.52 to 83.36. The removal of reversed bias significantly improves the performances of the task especially in the character level.

5 Contribution

In this project, we evaluate the state-of-the-art models ability to process jumbled sentences on three classic downstream tasks mimicking human cognitive abilities, including sentiment classification, information retrieval, and semantic similarity. When comparing the performance with 0.2 to 0.8 jumbling probabilities, we found the machines ability of reading jumbled sentences is more sensitive to the types of jumbling than degrees of jumbling, and the induced biases of jumbled embeddings greatly impairs performance. After visualizing the original and jumbled sentence embeddings, we find the possibility of removing induced bias to further improve the sentence embeddings accuracies in CR and TREC task and Pearson coefficient on the STS task. The removal of these induced biases significantly improves the machines robustness of reading character-level jumbled

Both authors of this paper contribute to the literature review, discussion of experimental design, and result analysis. Runzhe designs the jumbling schemes, implements the sentence embedding generation and evaluation pipelines for three tasks and experiments with the influence of types and degrees of jumbling on the performance of the jumbled sentence embeddings. Zhongqiao visualizes the jumbling and original sentence embeddings in t-SNE and PCA, and experiments with the effect of reversed bias on the performance of the jumbled sentence embeddings on all three tasks.

6 Conclusion and Future Work

In conclusion, when experiment with jumbling probability, we find out that the changing of the probability does not affect much on the performance except for the Omit operation on the word level. The Omit operation on the word level loses more information as we increase the probability. For the other schemes, they lose information with bigger probability but the under remaining information is still above machines extraction capacity. Therefore, the main restriction of performance is the machines capacity of capturing semantic information from jumbled sentence instead of competency or clearance of sentences.

We get significant improvement in all three tasks based on the induced bias removal, especially in character level. However, we are still not sure intuitively why the character level has such a clear and relatively stable bias when jumbling the text. We will further analysis the structure of the BERT model or the hidden effect of the character level jumbling in future, to find out which factors may contribute to induced bias and the improvement of the performance.

Our future work will continue to explore the jumbled sentence effect on more higher level language comprehension tasks, compare the machine performance variation with human performance variation, and use prior knowledge and context to improve the machine's robustness on processing jumbled texts without additional correctness and

annotation.

We also plan to conduct ablation study to analyze those methods to dig deeper into why they have different performances over handling jumbled text, what are the advantages and disadvantages for the design, what specific structures or component, e.g., language models (Peters et al., 2018), should help NLP systems to understand better, and finally how to improve based on those designs or a new design.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simlpe but tought-to-beat baseline for sentence embedding. *ICLR 2017*, page 16.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14.

Cheng Chang, Runzhe Yang, Lu Chen, Xiang Zhou, and Kai Yu. 2017. Affordable on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2200–2209.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Matthew H Davis. 2004. Units of representation in visual word recognition. *Proceedings of the National Academy of Sciences*, 101(41):14687–14688.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Spence Green, Sida I. Wang, Daniel M. Cer, and Christopher D. Manning. 2013. Fast and adaptive online training of feature-rich translation models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 311–321.

Thomas L Griffiths, Charles Kemp, and Joshua B Tenenbaum. 2008. Bayesian models of cognition.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Geoffrey E. Hinton and Sam T. Roweis. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 833–840.

Olof Jacobson and Hercules Dalianis. 2016. Applying deep learning on electronic health records in swedish to predict healthcare-associated infections. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2016, Berlin, Germany, August 12, 2016*, pages 191–195.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 67–72.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

K. Mayall, G. W. Humphreys, and A. Olson. 1997. Disruption to word or letter processing? The origins of case-mixing effects. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, 23(5):1275–1286.

Leo X. McCusker, Philip B. Gough, and Randolph G. Bias. 1981. Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception and Performance*, 7(3):538–551.

Shike Mei and Xiaojin Zhu. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence,*

January 25-30, 2015, Austin, Texas, USA., pages 2871–2877.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Graham Ernest Rawlinson. 1976. *The significance of letter position in word recognition*. Ph.D. thesis, University of Nottingham.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3281–3287.

Richard Socher, Victor Zhong, Caiming Xiong, and Sewon Min. 2018. Efficient and robust question answering from minimal context over documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1725–1735.

L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, and Sumit Negi. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data, AND 2009, Barcelona, Spain, July 23-24, 2009 (in conjunction with ICDAR 2009)*, pages 115–122.

Xiaobing Sun, Xiangyue Liu, Jiajun Hu, and Junwu Zhu. 2014. Empirical studies on the nlp techniques for source code data preprocessing. In *Proceedings of the 2014 3rd International Workshop on Evidential Assessment of Software Technologies*, pages 32–39. ACM.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

James T Townsend. 1990. Serial vs. parallel processing: Sometimes they look like tweedledum and tweedledee but they can (and should) be distinguished. *Psychological Science*, 1(1):46–54.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 90–94.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2016. Achieving human parity in conversational speech recognition. *CoRR*, abs/1610.05256.