Enhancing Indian Language Content on the Web

Sayar Ghosh Roy
Information Retrieval and Extraction Lab
Language Technologies Research Centre
International Institute of Information Technology, Hyderabad
sayar.ghosh@research.iiit.ac.in

Abstract

With the increase in number of individuals having a smartphone plus a strong internet connection in the past decade, the demand for online content has gone up greatly. In India, roughly half of the population using the internet are not fluent in English. Thus, there is a need to deliver coherent content in a person's language, dialect and a specific text style custom tailored for their demographic. India is a land of many tongues, and as such, we do not expect a one-size-fits-all solution to work out well in practice. Some other key features of Indian Languages (ILs) such as morphological attributes, free-word order, and pro-drop, make the modeling of ILs even more challenging. In this paper, we outline a few avenues to enhance the available content in Indian Languages (ILs) which is present on the web. Our suggestions are based on recent advances in Natural Language Processing and Information Retrieval. The approach is grounded in the concrete reality, is practical and scalable. Our design choices and decisions are as explainable as possible throughout.

1 Introduction

The amount of information available on the World Wide Web has gone up greatly in the past decade with an estimated number of a few million articles being published daily. A similar trend is seen for Indian Language articles as well. However, if we look at the relative content growth within specific ILs, we will find that this is not in proportion to the number of native speakers of these languages. From a time when about eighty percent of the web-pages on the internet were in English, we have come a long way to that percentage reaching the low thirties thereby increasing the representation of other languages. However, the amount of content in ILs such as Marathi, Malayalam, Kannada, Bhojpuri and many others is much lower than what we would expect.

There is a huge reader-base in India based on available statistics on the number of regional newspapers being sold and distributed across the country. The usage of smartphones and wireless internet has also increased greatly. However, only about two percent of the Indian population is fluent in English. Thus a need arises to provide quality content in regional languages to internet users. Apart from preservation and nurturing of individual native languages and traditions, it enables companies to reach a much wider audience base and have targeted advertisements which utilize some of the cognitive biases dependent on one's spoken language and dialect.

The ILs show a stark contrast among themselves when it comes to linguistic properties. India is home to languages from various language families, the most common ones being Indo-Aryan (which a child of Indo-European), Dravidian and Sino-Tibetan. Through extensive borrowing and nativization, we see similar sounding patterns across languages. For example, consider the retroflex 'r' sound which is now a part of Indo-Aryan languages like Hindi and Bengali but did not originate from the proto Indo-European class. Such instances of to and fro borrowing have provided some common features to ILs spoken in particular regions. But this is not enough make generalizable statements. In reality, the degree of differences not only among languages, but also among certain dialects causes us to model and handle each case differently. For example, consider the dialects of Bengali across various districts. As we move farther and farther from an origin, the spoken forms become more and more mutually unintelligible.

Thus, even though we use ILs as an umbrella term, there is no comprehensive way of grouping them into a single unit with an associated schema for modelling.

As a starting step, we focus on the standard or academic form of a particular language which is used in text and articles. We also focus on the twenty-two[1] official languages listed out in the Indian Constitution which is only a small fraction of all of India's spoken languages. Majority of ILs if not all, are considered to be resource-poor. We do not have enough parallel corpora for language translation and for most languages, large digitized corpora for modeling is also absent. We can leverage some of the known grammar rules for a language but the problem of law of diminishing returns creeps in with the usage of any rule based system. It is clear that we cannot rely on large parallel corpora which is available for Languages such as English, French, German or Spanish and need to focus largely on algorithms and architectures which are trainable in a non-supervised fashion.

We aim to enhance web content in Indian Languages. This involves translation of existing documents on the web into the target language, digitization of books, records and artefacts which are currently maintained on paper, generating custom content for individuals speaking the language explicitly taking their demographic into account. The second phase involving content-curation and conversion to digital formats needs to happen on a more localised scale and is primarily a matter of logistics and legalities. For the first and final phases, we try to automate certain procedures so as to generate text in specific target languages and styles, and populate web pages accordingly.

2 Problem

In this section, we formalize the task at hand. Given a document in any[2] language, we translate the text into the required target style. We define target styles as ordered 3-tuples of the form: (Channel, Age Group, Language). Channel describes the format of the document. Initially, we can support the following formats: a Facebook post, a Tweet, a blog post, a journal article, or a newspaper article. We segregate the age ranges into separate buckets as:

1. Children below 12
2. Children in the age group 12 to 18
3. Young adults in the age group 18 to 30
4. Mature adults in age range 30 to 45
5. Experienced adults in age group 45 to 65
6. The greatest generation aged above 65

For starters, we support the known International languages such as English, French, German, Chinese, Spanish and Portuguese and all the 22 official languages of India. The task involves understanding of particular styles of text and their attributes. It may involve selecting certain pieces of the text which are most suited to be framed within blog posts or Tweets and as such, it has the aspect of extractive text summarization (Mihalcea and Tarau, 2004; Singh et al., 2018; Cohn and Lapata, 2008) in addition to Machine Translation (Dabre et al., 2020; Yang et al., 2020; Garg and Agarwal, 2018).

3 Insights

We'll be using encoder-decoder (Bahdanau et al., 2014; Sutskever et al., 2014; See et al., 2017) architectures which we'll train and fine-tune with little to no parallel data. We expect our models to understand specific aspects about the different modes such as age group and channel in addition to just learning about the language formalism. We could start off by trying to classify the input text into a particular style tuple but in place of that, we adopt a 'multilingual' or rather language-independent input encoder which aims to understand the semantic content and informative attributes of the input text with a disregard to its particular style.

[1] Assamese, Bengali, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Sindhi, Tamil, Telugu, Urdu, Bodo, Santhali, Maithili and Dogri

[2] Within our defined set.

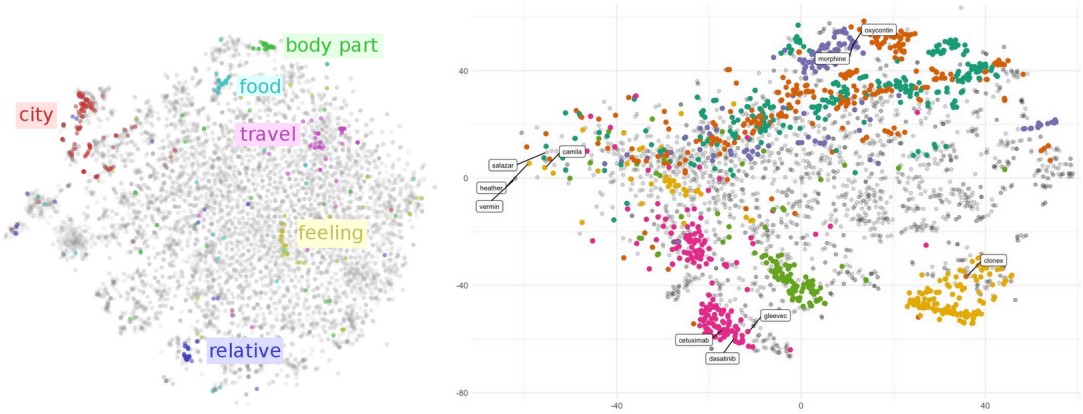

Figure 1: Embedding space. Left: General English, Right: Medical texts

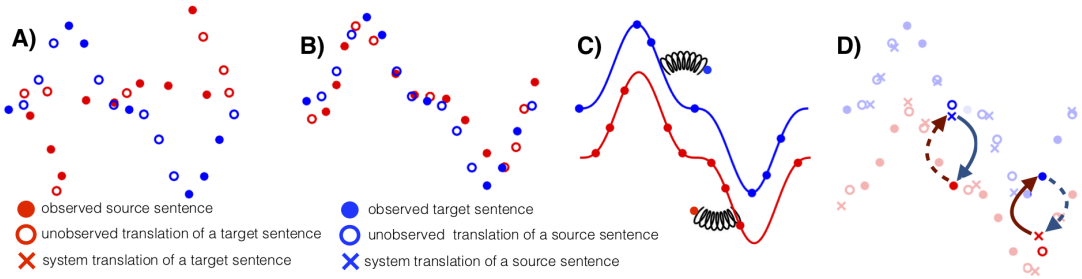

Figure 2: Neural Machine Translation without parallel corpora. (Lample et al., 2017)

Early work in this area was in the field of word translation without parallel data (Conneau et al., 2017). The key idea here is as follows: in different languages across the world, people largely talk about similar things and if we consider the view of word embeddings (Mikolov et al., 2013; Pennington et al., 2014) of every word in the vocabulary of a language in two dimensions, the shape of the embedding latent space is seemingly language independent. Figure 1 provides a view of word embeddings in different latent spaces: general domain independent corpora and medical science terms. Now, if we consider these shapes produced by embeddings in two different languages, they will differ slightly in orientation and size. If we can find a mapping function say F, which graphically translates the source language's embedding space to a new origin, rotates and scales it accordingly by certain factors followed by some internal manipulations, we could theoretically match the embedding space of the target language. A simple nearest neighbour based metric from source to target would then give us the word translation of the source word. The process is more involved and requires a loss function which tries to minimize the sum of distances between confident pairs of words (if the probability of a candidate target word for a chosen source is very high, that (source, target) duo is called a confident pair).

The successes in the above word translation task provided the necessary push to look into phrasal and sentence level translation without parallel data, which is scarcely available and expensive to create. Monolingual data on the other hand is readily available as parts of large corpora. A neural and phrase based model for language translation proved effective here (Lample et al., 2018a). The denoising effect of language models combined with the iterative back-translation procedure yielded results better than semi-supervised and fully-supervised approaches for low resource language pairs such as English-Urdu and English-Romanian. A very broad overview (see Figure 2) of the algorithm involved in translation without parallel data can be outlined as follows:

1. Building language models: Learn language models P_s and P_t over source and target languages
2. Initial translation models: Leveraging P_s and P_t, learn two initial translation models, one in each direction: $P^{(0)}_{s \to t}$ and $P^{(0)}_{t \to s}$

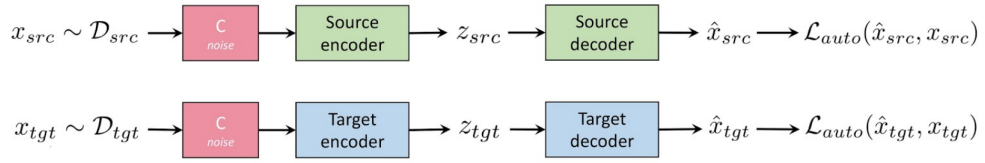

Figure 3: Denoising Auto-Encoding (Lample et al., 2017)

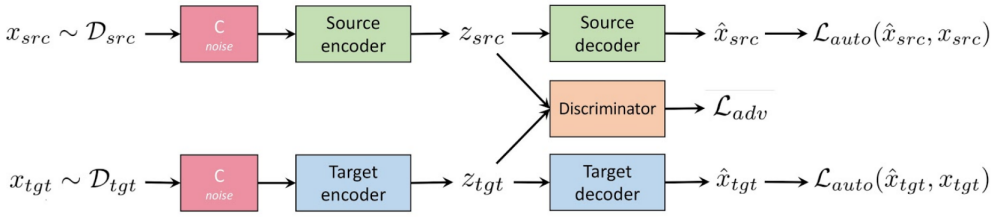

Figure 4: Adversarial Training (Lample et al., 2017)

3. For k = 1 to N, do

   (a) Back-translation: Generate source and target sentences using the current translation models $P_{t\to s}^{(k-1)}$ and $P_{s\to t}^{(k-1)}$, factoring in language models P_s and P_t

   (b) Train new translation models $P_{s\to t}^{(k)}$ and $P_{t\to s}^{(k)}$ using the generated sentences, leveraging P_s and P_t

For the auto-encoder training (see Figure 3), the added noise is of two kinds, namely, (a) Word dropout: each word is removed with a probability P (usually 0.1), and (b) Word shuffle: word order is (slightly) shuffled inside sentences. Auto-encoders are trained for both the source and target languages. The auto-encoder cells can comprise of Transformer (Vaswani et al., 2017) models or more traditional Recurrent Neural Networks (RNNs) using Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Chung et al., 2014)). After training the auto-encoder, source and target latent states are made indistinguishable using an adversarial training method (Yang et al., 2018). The idea here is to train a mapper and a discriminator model (see Figure 4). The job of the discriminator is to identify whether the text representation is from the source or the target side while the mapper tries to make things difficult for the discriminator by making the source and target spaces more alike in each iteration. Finally, the decoders need to operate in the same latent space which is achieved by sharing the parameters between source and target encoders.

In general, any task such as automatic text summarization (Tas and Kiyani, 2007; Nenkova and McKeown, 2012) or text style transfer (Fu et al., 2017; Li et al., 2018; Vadapalli et al., 2018; Lample et al., 2018b; Rao and Tetreault, 2018) can be viewed as a translation task. This is so because the overall goal is invariant: receive an input sequence and translate that into an output text with desired properties (Sutskever et al., 2014). Phrase based Statistical Machine Translation (PBSMT) was the main translation approach before neural techniques came into the picture. PBSMT is still a good candidate for unsupervised Machine Translation because it is based on memorization, has less parameters to fit and often beats neural methods especially when labeled data is scarce (low-resource language pairs). A PBSMT system requires a phrase table which is essentially a list of phrase pairs with direct and reverse translation probabilities. Seemingly, this component might require some parallel data but in recent literature, the approach for word translation discussed above has been applied upon phrases to automatically populate phrase tables based on mono-lingual corpora only.

Lastly, in recent works on neural text style transfer, transformer architectures have been leveraged to translate text from the style of one known author to another (Syed et al., 2020). Approaches using supervised data in such a setting have even looked at the problem of translating Shakespearean texts to

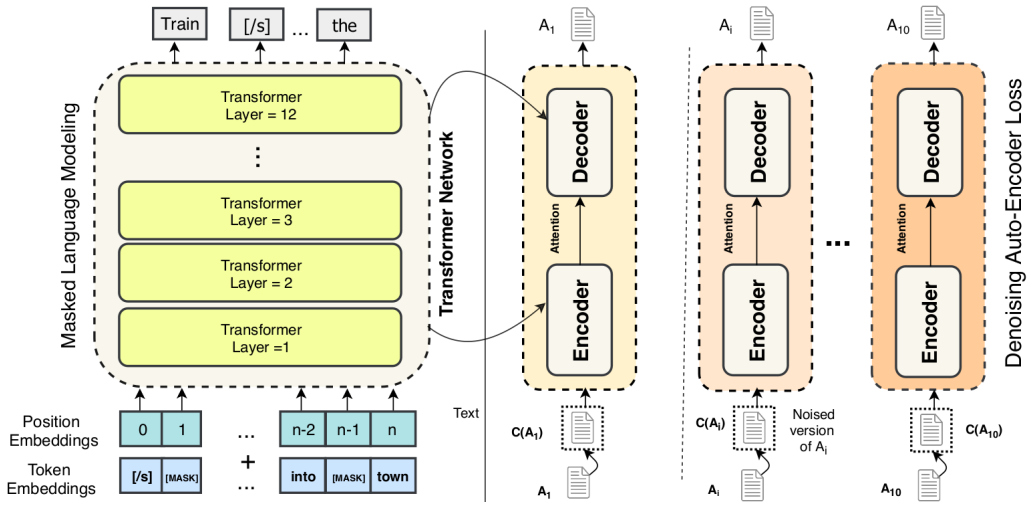

Figure 5: Transformer model overview (Syed et al., 2020)

plain English. One point to note here is that, on the whole, the ideas of having an encoder and decoder which can perform text to text translation remains unchanged (Raffel et al., 2019). In our approach, we plan to utilize multilingual Transformer models which are basically Transformer networks pre-trained with a Masked Language Model (MLM) pre-training objective followed up with specific fine-tuning objectives such the problem of Natural Language Inference (NLI). This is done across corpora in a wide variety of languages such that the source encoder maps the input into a latent space which is language independent. This idea is similar to the auto-encoder's desiderata in the problem of unsupervised machine translation. Thus, we aim to exploit the gains based on large scale language model pre-training (Devlin et al., 2018; Lewis et al., 2019) and utilize that to build the language to language and more specifically style to style translation system.

4 Approach

4.1 Data

In order to train our encoder-decoder models, a large amount of mono-lingual corpora will be required. Thus, a large scale scraping to find such content on the web is the very first step. Note that we do not need specific data for every possible configuration of our defined three-tuple. We only seek examples of each language, age-group style and platform-style such that our model can learn and generalize. For platform, we need to collect a set of news and journal articles, Tweets, Facebook posts, and blog posts. Similarly, for language, we can scrape Wikipedia pages written in the selected language in addition to certain web pages and news articles. Finally, for age group stylistics, we need a well defined mapping from age-group to potential websites which they frequent. For example, a website about war veterans from the 'Kargil War' will be tailored towards the greatest generation while a website on say, the cartoon character 'Ben Ten' will have content suitable for teenagers. This kind of a mapping might be expensive to create initially but will supply a great boost to our system's explainability and down the line performance.

We therefore create a scraping bot with associated URL handling rules to populate our URL field lists. We then use these URL sets to extract the boiler-piped contents from pages which serve as our corpora for the task. Moreover, these data-sets can be built incrementally and hence, good ablation studies based on the amount of data required for the system to reach a threshold score can be conducted in parallel. Owing to the support required from the scraped data, as future work, we can try a schema of federated learning where the incoming stream of URLs further aids in the incremental training.

4.2 Model and Training

The architecture is fairly straightforward and inspired by (Syed et al., 2020) (see Figure 5). We utilize a Transformer network in the unsupervised pre-training phase. Here, we experiment with: (a) A

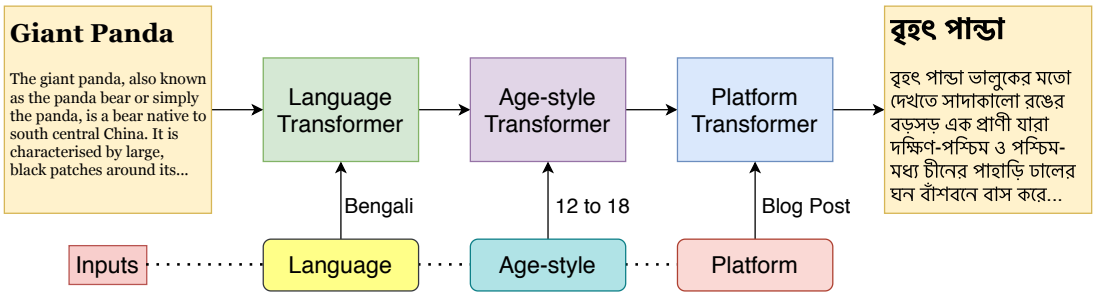

Figure 6: Proposed Model

simple twelve-layer transformer model, and (b) A pre-trained multilingual Transformer model such as mBERT (Reimers and Gurevych, 2020) or XLM-RoBERTa (Conneau et al., 2020). The transformer encoder network gets grounded to better represent and map input sequences to a unique latent space. This pre-trained encoder is used further for fine-tuning on specific languages, platforms, and age-group styles. Similar to work in unsupervised Machine Translation, parameters are shared across encoder and decoder models and we independently fine-tune to set model weights for each platform, age-group style as well as language.

Now, for the inference part, we go incrementally.(See Figure 6) Given an input text, the trained Transformer-based encoder-decoder model for the particular target language yields the necessary translated output. This is further passed into the model trained for the particular age-group. Intuitively, we expect this model to prune out certain words and phrases, replace it with more elderly-friendly jargon for more senior audiences, lower the Flesch-Reading Ease if the output is meant for children and so on. The final model for inference is the most abstractive one: the Transformer model changing the platform style. The Tweet generator should return a highly abstractive representation of a key thought with the potential to go viral, a news article should technically follow the pyramid structure of reporting while a blog-post should be more snappy and interesting.

We could run the inference in different orders as well and indeed, we can perform an ablation to see which manner of ordering works best: the proposed arrangement being the one which seems the most logical. It can be easily observed that the model is currently cascading into a three 24-layer Transformer for a full inference which is slightly expensive. Note that during fine-tuning, we are only training 24-layers at a time which is fairly achievable.

4.3 Evaluation

We train three multi-class classifiers for Language, Platform and Age. The collected data is used in a supervised setting and we train an encoder model, preferably another Transformer model (a bi-LSTM model should also work but the training time will be too high) with a multi-layer perceptron classifier head on top. These classifiers are run on the produced output from the final Transformer model responsible for the platform style shift and the results are tabulated during inference. A success corresponds to a perfect match of platform, language and age group. Other metrics based on Linguistic schemes have been proposed in the existing literature to evaluate such tasks but we stick to multiple classifiers as that would be more unbiased and reproducible.

5 Impact

The huge plus point in our approach is that it is highly data driven and strongly non-reliant on parallel corpora. Also, theoretically, we can handle all possible languages (using the simple Transformer network, a smaller set if we use pre-trained ones such as mBERT, XLM-RoBERTa, and DistilBERT (Sanh et al., 2020)). Given sufficient data, the model can scale easily to other domains, platforms, and potential styles and is highly generalizable. One can easily specify a different setting and a fourth fine-tuned decoder model can be prepared and appended to the chain for that purpose. Also, the model is more granular. In that, we can view the intermediates such as the raw language based Translation, or we could

choose to preserve the language and only alter the style. The possibilities are endless. The only drawback is the induced noise in the encoding & text generation cycle (Shah and Barber, 2018). We do not expect our models to perform perfectly at every stage and the noise produced in one translator, say for the age-group will percolate down to the next level as well.

The very generalizability of our approach can become a problem since it does not cater to different languages' specific modeling demands. Dravidian languages require rich modeling of the morphology since they use a great degree of derivational and inflexional affixes. Although, necessary pre-processing can be performed using state-of-the-art Morph analyzers[3]. In such a setting, structuretrans, has shown that one can pass additional features into a text-to-text Transformer network such as brackets for sentence structure and feature markers for part-of-speech. The possibilities for trying out various experiments by swapping out feature sets is limitless. For example, for morphologically rich languages, using chunk information, root word, affixes collection, and labels such as 'Is it the start of a sentence?' has been experimentally found to be beneficial (Agarwal et al.,). The problems specifically in enhancing content in Dravidian languages raise other questions regarding the input sequence itself and ordering of pre-processing tasks. Some design choices need to be made here such as: should we use a morph splitter first and have a sequence composed of root words and affixes, or should we include [root, affixes] as a single unit? The only drawback here becomes the requirement of supervised data. Labelling is often laborious, time consuming and expensive (in the sense that it requires trained annotators).

To make things more complicated, Indo-Aryan languages such as Hindi, Bengali, Marathi, and Oriya have free word order which makes sequence understanding and long range dependency modeling difficult. For example, in Hindi: 'raam ne khaana khaaya', 'khaana khaaya raam ne', 'khaana raam ne khaaya', 'khaaya raam ne khaana', 'khaaya khaana raam ne' (ram eat food, food eat ram, food ram eat, eat ram food, eat food ram) are all valid statements; their pragmatics might differ, but overall semantics remain the same. ILs do not have markers such as first word capitalization for marking of proper nouns. Foreign words, borrowed words, and the spelling norms for nativization are non-standardized. These create problems while recognizing certain patterns as the same sounding phrase may be spelt differently by different individuals. Indo Aryan Languages also have words formed by *Sandhi* and sandhi-splitting is a common pre-processing step while handling data from these languages. Almost all ILs show agglutination. Defining a universal data structure for storing agglutinative markers is a challenging task in itself. Last but definitely not the least, ILs are resource poor and finding good amounts of data especially for Sino-Tibetan languages like Nepali is particularly tricky.

6 Conclusion

In this system proposal paper, we looked at the problem of enhancing content in Indian Languages on the World Wide Web. We looked at specific pitfalls and drawbacks associated with such content creation and gained insights from the existing literature in the field. The lack of parallel data for effective model training caused us to look at architectures and algorithms which can work well by leveraging mono-lingual data as effectively as possible. We proposed a scheme to pre-train auto-encoder models followed up with further fine-tuning such that they can generate content for selected (platform, age-group style, language) settings. We looked at a way of curating URLs to have enough data for every possible language, platform or age group and proposed a scheme to continually generate content for specific audiences based on each input data-point. Finally, we looked at the potential downfalls of the approach and discussed a few underlying issues when it comes to processing Indian Languages in general. Our proposed system stands out as it can model a variety of settings with no supervised data and provides a great deal of versatility owing to its modular nature.

[3]Take it with a pinch of salt. Even the SOTA morph-analyzers for Dravidian languages do not always work out nicely in practice

References

Manish Agarwal, Rahul Goutam, Ashish Jain, Sruthilaya Reddy Kesidi, Prudhvi Kosaraju, Shashikant Muktyar, Bharat Ambati, and Rajeev Sangal. Comparative analysis of the performance of crf, hmm and maxent for part-of-speech tagging, chunking and named entity recognition for a morphologically rich language.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation.

Ankush Garg and Mayank Agarwal. 2018. Machine translation: A literature review.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018b. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

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)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*, 04.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

Harshil Shah and David Barber. 2018. Generative neural machine translation.

Abhishek Kumar Singh, Manish Gupta, and Vasudeva Varma. 2018. Unity in diversity: Learning distributed heterogeneous sentence representation for extractive summarization. In *AAAI*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting language models for non-parallel author-stylized rewriting. In *AAAI*, pages 9008–9015.

Oguzhan Tas and Farzad Kiyani. 2007. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213.

Raghuram Vadapalli, Bakhtiyar Syed, Nishant Prabhu, Balaji Vasan Srinivasan, and Vasudeva Varma. 2018. When science journalism meets artificial intelligence: An interactive demonstration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 163–168.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets.

Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. 2020. A survey of deep learning techniques for neural machine translation.