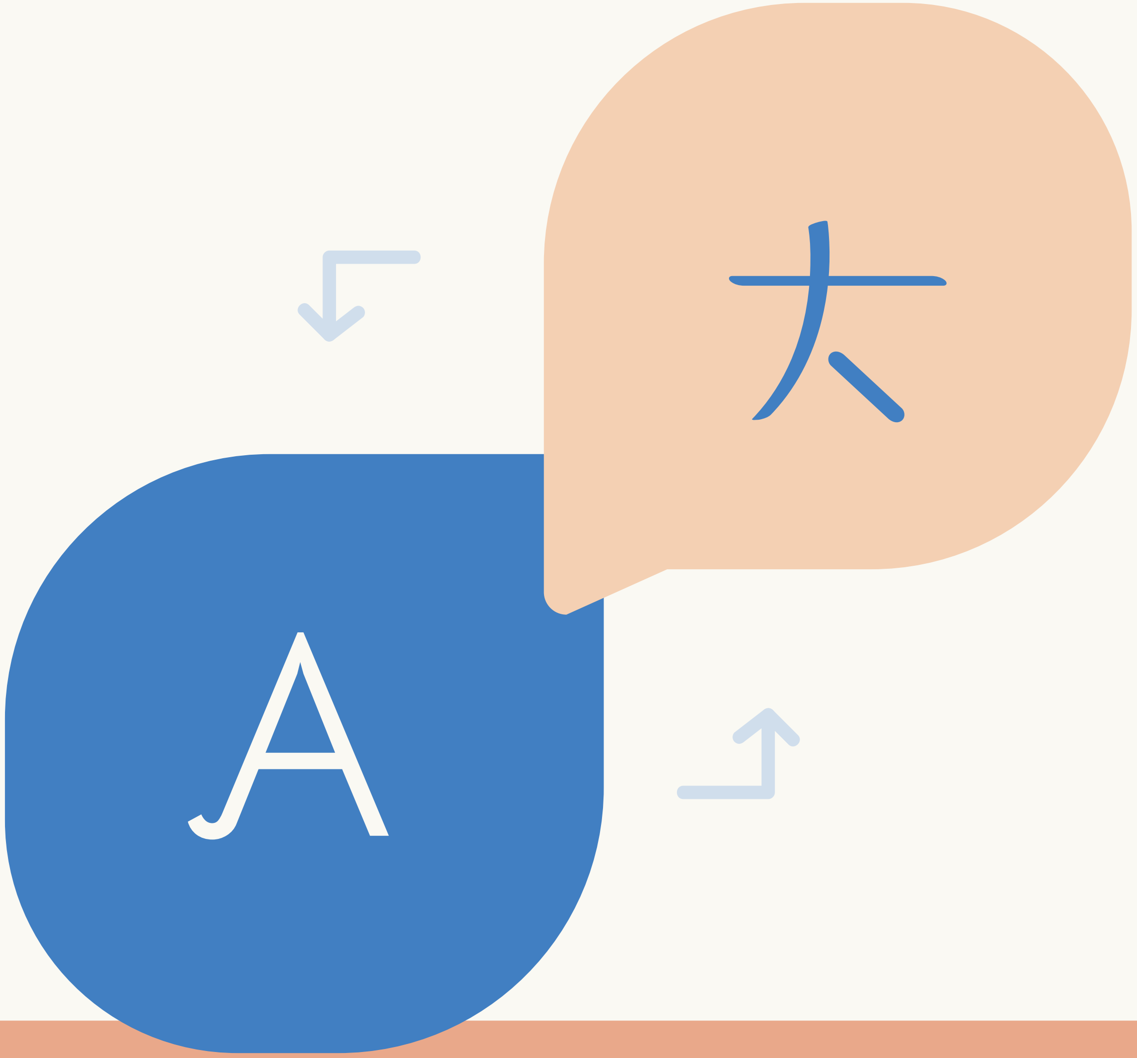


Journal of Computational & Applied Linguistics

Volume 1
2023

NEW
BULGARIAN
UNIVERSITY

Design and layout of edition
Janet Tsanova

Address
New Bulgarian University
Research Center for Computational
and Applied Linguistics
21, Montevideo Str., 309/ Building 2
1618 Sofia
jcal@nbu.bg

ISSN (online): 2815-4967

Journal of Computational & Applied Linguistics

Volume 1
2023

NEW
BULGARIAN
UNIVERSITY

Journal of Computational and Applied Linguistics (JCAL)

The Journal of Computational and Applied Linguistics **(JCAL) is a yearly open access peer-reviewed journal publishing articles which contribute new results in computational linguistics, applied linguistics and related areas. The goal of the journal is to bring together researchers and practitioners from academia and industry, to establish collaborations and to focus on advanced concepts in these fast-developing fields.**

The Journal has the following structure:

Articles
Reviews
Announcements
Bibliographies

Authors are invited to submit papers which illustrate research results, projects, surveying works and industrial experiences describing significant advances in the following fields:

Formal Grammars
Corpus Linguistics and Applications
Computational Linguistics
Contrastive Studies
Languages Technologies and Translation
Language for Specific Purposes
Multilingualism

Article Submission Guidelines:

File format: Microsoft DOCX;
Paper size: A4;
Orientation: Portrait;
Font: Book Antiqua;
Margins (Top, Bottom, Left, Right): 0.98 inch;
Line spacing: Single;
Paragraph style: Block format;
Title: block caps, bold, centred, size 14;
Author(s): Academic title, first name, last name, affiliation; font size 12, italics, right;
Abstract: font size 11;
Keywords: font size 10, italics;
Body of the article: font size 12, single line spacing, justified;
Spaces between paragraphs;
no header, footer or page numbers;
Footnotes: font size 10;
References: Harvard style, size 11, single line spacing.

Submissions must be original and should not have been published previously or be under consideration for publication while being evaluated for this Journal. They should be made via e-mail: jcal@nbu.bg.

Contents

6 **Text Data Augmentation Using Generative Adversarial Networks – A Systematic Review**
Kanishka Silva, Burcu Can, Raheem Sarwar, Frederic Blain, Ruslan Mitkov

39 **Extracting Algorithmic Complexity in Scientific Literature for Advance Searching**
Abu Bakar, Raheem Sarwar, Saeed-Ul Hassan, Raheel Nawaz

66 **Computer-assisted Transcription and Analysis of Bulgarian Child Speech Data using CHILDES and CLAN**
Velka Popova, Dimitar Popov

77 **Translation of Metaphors in Official and Automatic Subtitling and MT Evaluation**
Maral Shintemirova

94 **Gender-neutral Language Use in the Context of Gender Bias in Machine Translation (A Review Literature)**
Aida Kostikova

reviews

110 **How Much Linguistics in Corpus Linguistics? Review of *Doing Linguistics with a Corpus* by Egbert, J., Larsson, T. and Biber, D. (2020). Cambridge University Press**
Maria Stambolieva

https://doi.org/10.33919/JCAL.23.1.1

TEXT DATA AUGMENTATION USING GENERATIVE ADVERSARIAL NETWORKS – A SYSTEMATIC REVIEW

Kanishka Silva[1*], Burcu Can[2], Raheem Sarwar[3], Frederic Blain[4], Ruslan Mitkov[1]

[1] University of Wolverhampton, Wolverhampton, United Kingdom

[2] Department of Computing Science and Mathematics, University of Stirling, Stirling, United Kingdom

[3] Department of Operations, Technology, Events and Hospitality Management, Faculty of Business and Law, Manchester Metropolitan University, United Kingdom

[4] Department of Cognitive Science and Artificial Intelligence, Tilburg University, Tilburg, The Netherlands

[*] Corresponding author. Email: kanishka.silva@wlv.ac.uk

Abstract

Insufficient data is one of the main drawbacks in natural language processing tasks, and the most prevalent solution is to collect a decent amount of data that will be enough for the optimisation of the model. However, recent research directions are strategically moving towards increasing training examples due to the nature of the data-hungry neural models. Data augmentation is an emerging area that aims to ensure the diversity of data without attempting to collect new data exclusively to boost a model's performance.

Limitations in data augmentation, especially for textual data, are mainly due to the nature of language data, which is precisely discrete. Generative Adversarial Networks (GANs) were initially introduced for computer vision applications, aiming to generate highly realistic images by learning the image representations. Recent research has focused on using GANs for text generation and augmentation. This systematic review aims to present the theoretical background of GANs and their use for text augmentation alongside a systematic review of recent textual data augmentation applications such as sentiment analysis, low resource language generation, hate speech detection and fraud review analysis. Further, a notion of challenges in current research and future directions of GAN-based text augmentation are discussed in this paper to pave the way for researchers especially working on low-text resources.

Keywords: *Text Data Augmentation, Generative Adversarial Networks, Adversarial Training, Text Generation*

1. Introduction

Computational models in deep learning and machine learning usually perform better when high-quality and balanced datasets are available in natural language processing applications. However, it is usually challenging to obtain a high-quality dataset; for instance, in supervised learning tasks, we often need to deal with the lack of labelled data or a limited amount of labelled data, which directly affects the model's performance. Obtaining a large-scale dataset is time-consuming and associated with a higher cost. Therefore, expanding a given smaller dataset artificially for any natural language processing task is a promising solution. Applying data augmentation for NLP tasks, specifically for text-based applications, may exhibit lower accuracies due to language-variant characteristics such as grammatical structure. For instance, according to Luo et al. (2021), a text classification task would fail to improve performance due to grammatical errors or uncontrolled sentiment characteristics in the generated text. Although we need more data in data augmentation, replicating data is not a solution, as it will eventually lead to model overfitting.

Generative Adversarial Networks (Goodfellow et al., 2014) aim to synthesise real-world data as closely as possible. As improvements to the original GAN model proposed by Goodfellow et al., several other studies stabilised GAN training along with different loss functions (Nowozin et al., 2016; Mao et al., 2017; Arjovsky and Bottou 2017). Several other notable GAN architectures are Conditional Generative Adversarial Networks (Mirza and Osindero, 2014), Deep Convolutional Generative Adversarial Networks (Radford et al., 2018), Coupled Generative Adversarial Networks (Liu and Tuzel, 2016), Cycle-Consistent Generative Adversarial Networks (Zhu et al., 2017) and Information Maximizing Generative Adversarial Networks (Chen et al., 2016). Given the objective of GAN models, generating new data while being closer to the original data distribution is feasible to apply for data augmentation.

This paper aims to pave the way for researchers especially working on low textual resources, by reviewing previous work in textual data augmentation using GAN models in various NLP application domains. In this sense, this paper is the first systematic review focusing on GAN-based text data augmentation. Furthermore, we surveyed text augmentation application domains such as sentiment analysis, hate speech detection, low resource language generation and fraud text identification.

The research questions for this systematic study are as follows:

1. How can text augmentation help to improve a computational model's performance?
2. How can GAN models be utilised for text data augmentation?
3. What are the challenges in GAN-based text augmentation worth addressing in future research?

The rest of the paper is structured as follows: Section 2 describes the methodology followed for the systematic review and paper screening, such as inclusion and exclusion criteria. Section 3 briefly introduces data augmentation, and Section 4 presents a comprehensive overview of Generative Adversarial Networks. Section 5 systematically reviews a few applications using GAN-based text augmentation. Section 6 summarises text data augmentation challenges and potential future directions. Finally, Section 7 summarises the objectives of this study.

2. Methods

This systematic review adheres to Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Moher et al., 2009). We filtered the articles through a well-defined inclusion-exclusion strategy per the PRISMA guidelines following through the identification, screening, exclusion, and inclusion stages. Figure 2.1 shows the PRISMA flowchart we used with filtered paper counts in each stage.

We conducted the search initialisation as per the PRISMA guidelines (Moher et al., 2009) and collected articles from digital libraries such as Scopus, Web of Science, IEEE Xplore, Science Direct, Google Scholar and Semantic Scholar, which were published between 2017 and 2022, with a search duration spanning from March 2022 to May 2022. We used some keywords to search the databases. Initially, we used key phrases such as "text data augmentation using generative adversarial networks" and "text augmentation using GAN". We then narrowed the search to the scope of applications, such as "Generative Adversarial Network data augmentation for fraud text identification" and "low resource language generation using GANs". Further, we utilised complex search strings to combine similar keywords with AND and different keywords with OR. For instance, "text augmentation" AND "text synthesis" and "text augmentation for low resource languages" OR "synthesised text in semantic analysis". Altogether we collected 257 papers initially and removed 96 duplicate entries, resulting in 161 papers for the screening stage.

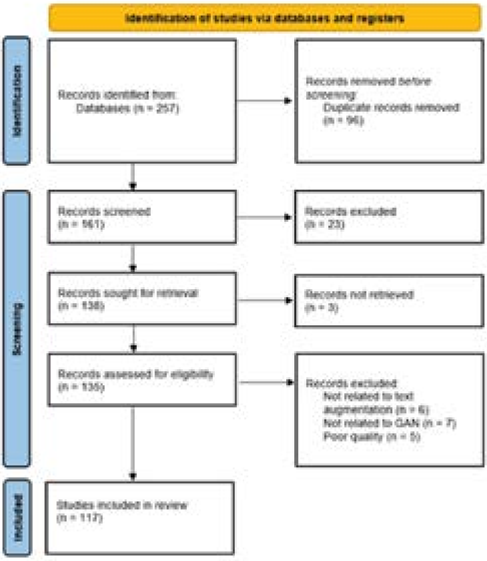

Identification of studies via databases and registers
Identification
Records identified from:
Databases (n = 257)
Records removed before screening:
Duplicate records removed (n = 96)
Screening
Records screened (n = 161)
Records excluded (n = 23)
Records sought for retrieval (n = 138)
Records not retrieved (n = 3)
Records assessed for eligibility (n = 135)
Records excluded:
Not related to text augmentation (n = 6)
Not related to GAN (n = 7)
Poor quality (n = 5)
Included
Studies included in review (n = 117)

Figure 2.1: The PRISMA guideline flowchart used in this review (Moher et al., 2009)

Twenty-three articles were excluded during the screening process upon careful scan through the title and abstract. Then another exclusion step was performed considering full-text availability, which excluded three papers from the results. In the final step in screening, we considered whether the selected papers aligned with the stated research questions. We excluded 17 papers since they were unrelated to text augmentation or GAN, and some had poor-quality content. A total of 117 articles were selected eventually, and the distribution is illustrated in Figure 2.2. Finally, the papers were grouped hierarchically for a clear presentation in the review. Several papers were included during the write-up period since those papers were vital in explaining the theoretical background.

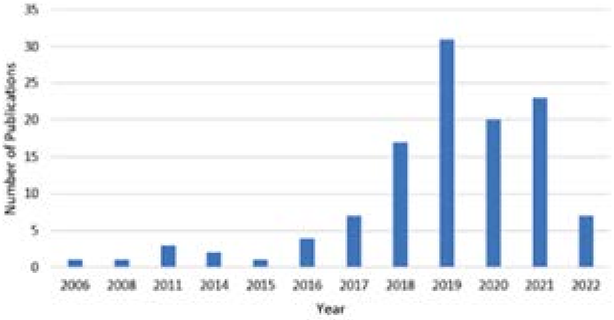


Figure 2.2: Numbers of selected publications over the years

3. Data Augmentation

Data augmentation generates a massive amount of data from a given small set of available data, guaranteeing an increased model accuracy. The simplicity of the proposed data augmentation approaches is a must to replace the time-intensive and cost-ineffective manual data collection and annotation to increase the size of an existing small-scale dataset. Feng et al. (2021) claim that a simple augmentation approach and accuracy boosting are trade-offs in data augmentation because overfitting will occur if the generated data is too identical to the original one. Therefore, the augmented data should be similar but deviate from the original data distribution. A typical approach is to perform data augmentation before the training is conducted and then mix the augmented data with the existing training data for training purposes. Another approach is generating data while the training occurs, a common technique in GAN-based data augmentation, especially in computer vision applications.

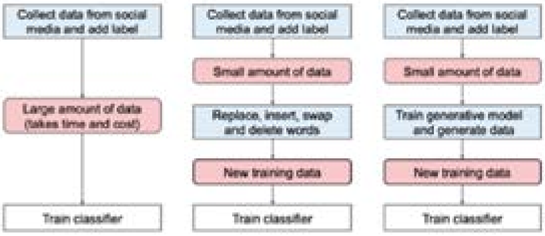

Collect data from social media and add label
Large amount of data (takes time and cost)
Train classifier
Collect data from social media and add label
Small amount of data
Replace, insert, swap and delete words
New training data
Train classifier
Collect data from social media and add label
Small amount of data
Train generative model and generate data
New training data
Train classifier

Figure 3.1: The methods used for collecting training data for a classifier. Left to right: a general method, dictionary-based data augmentation, generative model-based data augmentation (Luo et al., 2021)

Recent trends in NLP applications are heading towards leveraging large pre-trained models, especially in low-resource domains. Due to the exploration of new tasks, more data is the primary demand, but it is costly and time-intensive to annotate a large set of training data manually. Since high-quality data ensures the model's accuracy in conventional NLP approaches, it is difficult to turn a blind eye to this research gap. Moreover, low-resource scenarios, such as low-resource language data generation, also require a decent amount of training data. In such cases, augmenting data artificially is quite reasonable and adequate.

Overall, three techniques are used in data augmentation rule-based, example-interpolation-based and model-based (Feng et al., 2021). Rule-based approaches either consider the model's feature space (Xie et al., 2020; Wei and Zou 2019; Paschali et al., 2019) or use a graphical representation of the individual sentences (Chen et al., 2020; Şahin and Steedman, 2018). The example-interpolation technique takes two or more real examples and then alters the input and output labels. MIXUP architecture (Zhang et al., 2018) which follows the example-interpolation technique, has been later developed into different variations. Such variations are CUTMIX (Yun et al., 2019), which mixes two selected example images by replacing small sub-regions and Seq2MIXUP (Guo 2020), which generalises MIXUP for the sequence transduction task. Model-based techniques use sequence-to-sequence (seq2seq) models (Kumar et al., 2019; Sennrich et al., 2016) and language models based on recurrent neural networks and transformers (Sennrich et al., 2016; Yang et al., 2020).

Several data augmentation approaches in NLP include facilitating low-resource languages such as Turkish, Nepali, and Sinhala (Fadaee et al., 2017; Qin et al., 2021), bias mitigation (Zhao et al., 2018; Lu et al., 2020) and adversarial training (Jia et al., 2019; Kang et al., 2018). Moreover, applied NLP tasks that use data augmentation for performance gain involve classification (Wei and Zou 2019; Chen et al., 2020; Anaby-Tavor et al., 2020), summarisation (Fabbri et al., 2021; Parida and Motlicek 2019; Zhu et al., 2022), question answering (Longpre et al., 2019; Yang et al., 2019; Riabi et al., 2021), and dialogue systems (Quan and Xiong 2019; Louvan and Magnini 2020; Hou et al., 2018; Kim et al., 2019).

Initial approaches in textual data augmentation involve replacing words with synonyms or removing random words (Wei and Zou, 2019), which is not promising because of minor accuracy improvements due to overfitting, mainly in classification tasks. The data augmentation strategies followed for the textual data fall into three main categories: dictionary-based data augmentation, generative model-based, and general method, as in Figure 3.1 (Luo et al., 2021). Wei and Zou (2019) proposed a data augmentation strategy for text classification using a synonym dictionary to randomly increase the number of data points by inserting, replacing, deleting and swapping a word in a sentence. However, the performance with the synonym dictionary method (Wei and Zou, 2019) drops when the original data changes by more than a 10% ratio. Such approaches often exhibit the limitation of retaining sentiment information and even result in a drastic change in the actual sentiment class (Luo et al., 2021).

Generative models align with the probability distribution of the training data upon new data generation. Given that text generation is a complex task, such approaches were not entirely promising in text-based applications, specifically in classification models (Luo et al., 2021). Several generative models based on data augmentation were proposed by Anaby-Tavor et al. (2020), Feng et al. (2020), Radford et al. (2019). Apart from these text-generation strategies for text augmentation, generative adversarial networks are gaining popularity due to generating similar but fake data. Most data augmentation applications using GANs are in the computer vision area. However, there has been an increasing interest in using GANs for text data augmentation in the last few years.

4. Generative Adversarial Networks (GANs)

Machine learning models can be categorised into generative models and discriminative models. The discriminative models involve classification tasks that aim to predict the class labels by modelling a given feature set of inputs. In generative models, given the class and introduced noise, the distribution of the feature set is generated. Goodfellow et al. (2014) introduced a powerful generative model, Generative Adversarial Networks (GANs), adhering to a minimax game of two competing networks. The GAN model's main components compose a generator similar to a decoder and a discriminator that functions as a classifier. GANs have produced high-quality and diverse images for data augmentation in computer vision applications. Several GAN models which address image data are: face generation using StyleGAN (Karras et al., 2019), image translation using CycleGAN (Zhu et al., 2017), transforming doodles into pictures using GauGAN (Park et al., 2019) and generating 3D images using 3D-GAN (Wu et al., 2016), Wasserstein-GAN (Arjovsky and Bottou, 2017), coupled-GAN (Liu and Tuzel, 2016) and StackGAN (Zhang et al., 2017). The underpinning theories with these GAN applications deviate from the text data generation using GANs in minor aspects, but the intuition is the same by adhering to generator-discriminator architecture.

In GAN architecture, generator G learns to create fake samples that resemble real examples, and discriminator D learns to distinguish real samples from fake samples. The generator model is not sophisticated at the beginning to allow stable training. The discriminator mimics a classifier's behaviour. The probability outputs generated by the discriminator serve as an input for the generator. Both generator and discriminator are based on two separate neural networks. Figure 4.1 illustrates a GAN architecture. The input to the generator model is random noise, and the outputs are also randomly generated noisy samples. The generator expects to be as primitive as possible at this stage. Then the output is tuned with the response obtained from the discriminator. The generated samples become closer to the original data instances as the training continues. Following a minimax game theory, the generator and discriminator act as opponents trying to fool each other, eventually increasing the GAN model's performance on a particular task. The discriminator takes both original samples and the feature distribution of generated fake samples to classify both samples. Finally, when the discriminator cannot perform the classification correctly anymore, it is the point where the generator starts to make new samples which do not exist in the training data. Applications of GANs include super-resolution, assisting artists and element abstraction, specifically in the image domain.

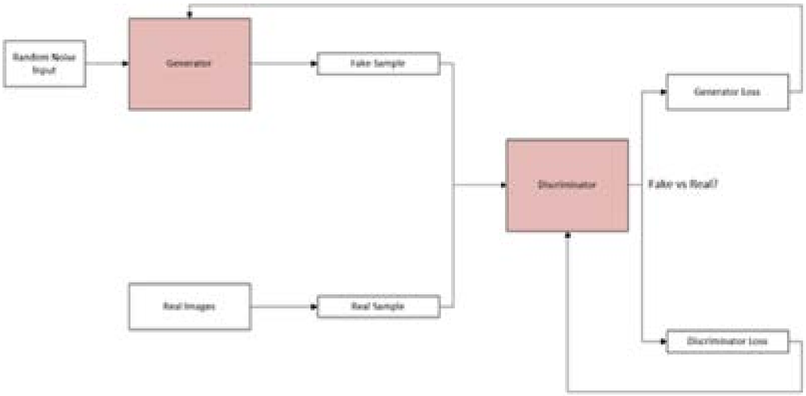


Figure 4.1: GAN Architecture

GAN models use adversarial concepts of producing fake samples mimicking real ones. The overall model improves continuously until an equilibrium point is reached due to competitive training of both the generator and discriminator. This concept is called the Minimax game, a decision rule with alternate moves for both players. Only one player wins by maximising their win in this concept, while the other tries to minimise the loss. Borrowing this idea for the GAN model, the generator tries to minimise the probability output of the discriminator, which is labelled as 'fake'. Simultaneously, the generator maximises the probability of classifying real and fake samples.

Equation (1) mathematically defines the minimax game of a GAN model: G is the generator, D is the discriminator, x denotes the real sample input, and D(x) is the probability of the label for the real sample. While z is the noise or the latent space vector used to provide inputs to the generator, G(z) indicates generated fake samples. The discriminator outputs that are expected for these two classes, respectively, are G(x) = 1 and D(G(z)) = 0. Mainly, the objective of the generator is to make the discriminator identify fake samples as real ones, i.e., D(G(z)) = 1, which results in minimising 1-D(G(z)):

$$\min_{G} \max_{D} V(D,G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[1 - \log D(G(z))] \tag{1}$$

When training the generator to minimise 1-D(G(z)), the generator's output should collectively provide input to the discriminator. Then the discriminator's loss should be backpropagated into the generator. To pass the loss gradients back to the generator, the selection criteria within the generator should be a differentiable function.

If we consider an RNN-based text generator, the next word in a sentence generated at each time step corresponds to the one with maximum probability in the softmax distribution. Suppose the GAN generator is implemented using a similar RNN to generate texts. However, the corresponding picking function is non-differentiable in the GAN generator. This issue does not apply to continuous data such as images. Using GANs for text generation is challenging due to the nature of textual data, which does not involve continuous and numerical data. However, since the text does not carry any of these features, despite the challenges, the following approaches were introduced to utilise GANs for text generation: the reinforcement algorithm-based method (Yu et al., 2017), the Gumbel-softmax approximation method (Kusner and Hernández-Lobato, 2016) and the method of avoiding discrete spaces (Donahue and Rumshisky, 2018).

Using reinforcement learning is presented by Fedus et al. (2018) and Yu et al. (2017). Suppose text generation is performed via a Reinforcement Learning (RL) agent, where the agent generates the next word based on the current state s, the previously generated sentence. A word vocabulary is used to define the action set. A reward is received once the RL agent reaches the end of the sentence action. In GAN architecture, the discriminator returns the overall reward.

Given the start state S_0, ϕ-parameterised discriminator model $D_ϕ$, sequence to produce $Y_{1:T} = (y_1, \dots , y_t, \dots ,y_T)$, current state $s = Y_{1:t-1}$ and the reward for a complete sentence RT, the θ-parameterised generator model G_θ, a gradient method is utilised to find the optimal parameters θ^* by applying gradient descent as follows:

$$\theta \leftarrow \theta + \alpha_h \nabla_\theta J(\theta) \qquad (2)$$

while maximising the overall reward as given below:

$$J(\theta) = \sum_{y_1 \epsilon Y} G_\theta(y_1|s_0) Q_{D\phi}^{G\theta}(s_0, y_1) \qquad (3)$$

A discriminator network performs classification on input sentences by providing a metric of how real it is. G represents parametrised policy $\pi(a|s,\theta)$ which takes a set of words as input to produce a probability distribution for the next word. During the training process, Monte-Carlo rollouts calculate an intermediate reward, and the discriminator provides the reward for the entire sentence. Persisting issues with this method include high variance in gradient estimate with each episode, resulting in an unstable training process and slow convergence. Pretrained generator and discriminator models can speed up training to solve these problems. Another problem also occurs when the state-action space is vast; for example, with an extensive vocabulary set, it tends to converge to local minima.

Due to the issues mentioned earlier with the Reinforcement Learning approach, recent research focuses on investigating other solutions for discrete data generation using GAN models. Selecting the next word in text generation maximises the probability generated via the softmax function at each time step. This selection operation is non-differentiable. Suppose the output y is a one-hot-vector with |V|-dimensions and h hidden states. Then the sampling is performed as follows:

$$p = softmax(h) \qquad (4)$$

Another sampling method is to use a vector of samples g from a Gumbel distribution as follows:

$$y = one\_hot(arg\, max_i(h_i + g_i) \qquad (5)$$

To make the argmax() function differentiable, a softmax approximation and an additional temperature parameter τ are introduced as given below:

$$y = softmax(1/\tau(h+g)) \qquad (6)$$

so that when $\tau \rightarrow 0$, the output distribution converges to a one-hot vector. During the training, τ is initialised with larger values, which converge on zero, as mentioned in Kusner and Hernández-Lobato (2016) and Donahue and Rumshisky (2018).

In encoder-decoder mapping, the encoder projects the input space onto a smaller dimensionality, and the decoder reconstructs the input from this representation. The solution for GAN text generation is not to consider it a separate discrete token generation. Instead of decomposing a given input sequence of discrete word tokens, this approach works with continuous space vectors, which are not human-readable. The problem arises in the discriminator's input representation while feeding the real sentences, which the auto-encoder facilitates. At the end of the training, the generator network outputs sentence vectors.

5. GAN for Text Data Augmentation

GANs have already been used for text data augmentation for various NLP applications listed below. However, before reviewing such NLP applications, it is noteworthy to mention GAN models' drawbacks in classification tasks such as sentiment analysis. For example, GANs may generate augmented data in opposite polarity, drastically impacting a sentiment analysis task. Nevertheless, GAN-based data augmentation can mitigate class imbalance problems by generating missing class data with controlled generation. Moreover, in the tasks such as bot-generated data identification, GAN-based fake data generation provides a promising adversarial approach. Collecting and analysing such datasets manually in practical cases is difficult.

5.1 Applications

Many NLP applications have used GANs for text data augmentation. These NLP applications include sentiment analysis, hate speech detection, low resource language generation, fraud detection, and code-switching sentence generation.

5.1.1 Sentiment Analysis

The challenges in sentiment analysis include a lack of data for low-resource languages and an imbalance issue in available datasets. Transfer learning (Gupta et al., 2018) and semi-supervised learning (Goldberg and Zhu, 2006) are alternatives in low-resource scenarios, but text-generation models also facilitate such problems. As mentioned in (Gupta, 2019), several techniques were introduced for sentiment analysis in low-resource scenarios, such as semi-supervised learning (Socher et al., 2011), regularisation methods (Gupta et al., 2018; Sindhwani and Melville, 2008) and latent variable models (Täckström and McDonald, 2011).

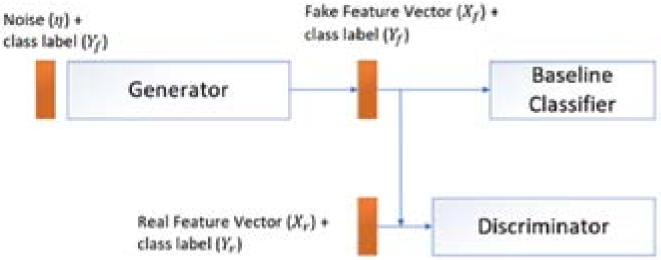

Noise (η) + class label (Y_f)
Generator
Fake Feature Vector (X_f) + class label (Y_f)
Baseline Classifier
Real Feature Vector (X_r) + class label (Y_r)
Discriminator

Figure 5.1: cGAN architecture (Gupta, 2019)

A variation of conditional GAN for low-resource datasets was introduced by Gupta (2019) with a baseline classifier in place apart from the generator and discriminator model. The implementation follows three approaches to ensure convergence: model pretraining from an available large dataset, input noise addition, and one-sided label smoothing, as illustrated in Figure 5.1. Both generator and discriminator employ feed-forward neural networks. The baseline classifier is pre-trained on a target task dataset and uses a shallow neural network architecture. The cross-entropy loss is used to learn the discriminator parameters as follows:

$$L_D = -y\log(D([x_r;y_r])) - (1-y)log(1-D([x_f;y_f])) \quad (7)$$

Here, each $[x_f; y_f]$ represents the concatenation with a label representation yf while assigned probabilities at discriminator are denoted by $D([x_r;y_r])$ and $D([x_f;y_f])$. Two generator losses are combined as given below:

$$L_G = L_{G1} + \lambda L_{G2} \quad (8)$$

where $L_{G1} = -\log(D([x_f; y_f]))$; $xf = G(\eta)$; $L_{G2} = -CE(y_f, C(x_f))$ (9)

The standard generator loss L_{G1} is to fool the discriminator while L_{G2} is to handle cross-entropy loss on the base classifier with λ hyper-parameter. $G_{(\eta)}$ corresponds to the generated output x_f with noise input η (Gupta, 2019).

Evaluation in Gupta (2019) is performed on the base classifier C_b, cGAN classifier C_f and a classifier on Twitter data C_t. Due to the discriminative power of generated data, C_f performs better, and the accuracy of C_t is mainly due to knowledge transfer. The evaluation of movie and product reviews has shown a significant accuracy increase of 1.76% and 1.7%, respectively, compared to the base classifier, which only uses actual data without utilising the generated data. As shown in Figure 5.2, T-SNE distribution and the projection of real vs fake data reveal that the generated data does not cover real data's entire feature space. Further, it is not easy to find a massive pre-trained dataset for the data augmentation task. Future directions include selective data generation in smaller spaces.

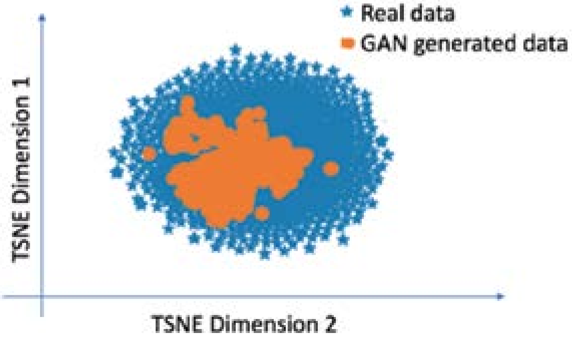


Figure 5.2: Real and fake data distribution, as observed on a 2-D projection of data points obtained using the t-SNE method (Gupta, 2019)

Another issue in sentiment analysis is the training on long texts in a low-resource dataset. As mentioned before, text generation models are prone to generating inaccurate sentiment information for the generated texts. Luo et al. (2021) propose a penalty-based SeqGAN for generating high-quality long-text data improving the SeqGAN model (Yu et al., 2017). The main challenge in using long text data is the low accuracy obtained when using such long text data in a classifier. The works of Luo et al. (2021) present an LSTM model with attention which performs sentence compression for the given training data. A sentiment dictionary aids in addressing the issue of losing sentiment words during the compression. With RL to address discrete data issues, the generator produces sentence sequence s based on the x token of the real word. The GAN model consists of a parameterised generator $G(\theta_g)$ and a discriminator $D(\theta_d)$ that aim to maximise the reward $G(x|s;\theta_g)D(x;\theta_d)$:

$$J_G(x) = \begin{cases} \mathbb{E}_{x\sim P_g}[-\log(D(x;\theta_d))] \\ \mathbb{E}_{x\sim P_g}[-\log(G(x|s;\theta_g)D(x;\theta_d))] \\ \mathbb{E}_{x\sim P_g}[G(x|s;\theta_g)V(x)] \end{cases} \quad (10)$$

The applied penalty-based objective on the generator is forced to minimise the overall penalty $\mathrm{G}(x|s;\theta_g)V(x)$ given that $V(x) = 1 - D(x;\theta_d)$, which leads to generating grammatically correct sentences.

Compared to the previous cGAN model (Gupta, 2019), this model requires no pre-training step with another dataset on the target task. The evaluation parameters involve classification accuracy, usability, novelty, and the diversity of the generated data, which outperforms the state-of-the-art accuracy (Wei and Zou, 2019).

5.1.2 Hate Speech Detection

Hate speech detection is usually performed by supervised models. However, most of the available datasets are imbalanced, which is one reason for the low performance of the hate detection models. Applying data augmentation for the class with fewer examples is a reasonable solution, but this is a challenging task for text generation. Cao and Lee (2020) introduce HateGAN, a GAN model aiming for hate speech detection using a deep generative RL model based on hateful tweets. The overall architecture is illustrated in Figure 5.3. The model adopts SeqGAN (Yu et al., 2017) by adding a toxicity scorer (Figure 5.4), which is pre-trained as a multi-label classifier to provide realistic scores and hate scores.

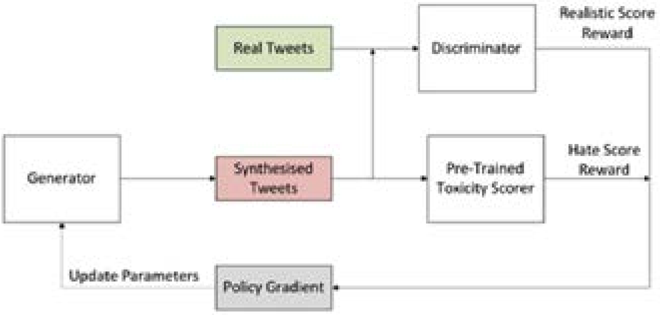


Figure 5.3 Architecture of the HateGAN model (Cao and Lee, 2020)

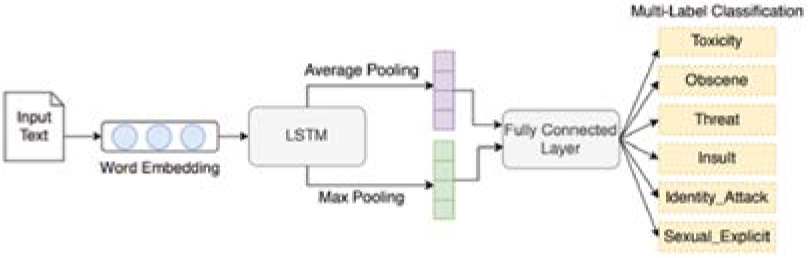


Figure 5.4: Toxicity scorer that is pre-trained as a multi-label classification model (Cao and Lee, 2020)

Given that S is a scoring module, N is the number of Monte Carlo searches, and x_i is the i-th Monte Carlo result, the expected reward from a sentence which is an action value for selecting the t-th word w_t is computed as follows:

$$r(state = (w_1, \dots, w_{t-1}), actions = w_t) = \frac{1}{N}\sum_{i=1}^{N}(S(x_i)) \quad (11)$$

The loss as a negative expected reward is defined as follows:

$$\begin{aligned} Loss(\alpha) &= -\sum_{t=1}^{n} \mathbb{E}_{[w_{1:t-1}]\sim G_\alpha}[\mathbb{E}_{w_t\sim G_\alpha}[r(w_t)]] \\ &\approx -\sum_{t=1}^{n} \sum_{w_t\in V} G_\alpha(w_t|w_{1:t-1}) \frac{1}{N} \sum_{i=1}^{N} S(x_i^t) \end{aligned} \tag{12}$$

The final combined reward becomes:

$$r(x) = Discriminator(x) + \sigma ToxicityScorer(x) \tag{13}$$

where x is the input sentence and σ is a hyperparameter.

5.1.3 Low Resource Language Generation

Question Answering (QA) is useful in deep learning since many deep learning applications can be modelled as QA problems. Developing a QA system in a low-resource language is challenging due to insufficient annotated datasets. For instance, according to Sun et al. (2019), a low-resource language, Tibetan demonstrates challenges in building such a question-answering model because of the language features such as longer sentences, complex syntactic structures and strict grammatical rules. Sun et al. (2019) introduce QuGAN, using Quasi-Recurrent Neural Networks (QRNN) and Reinforcement Learning as a QA corpus generation model for the Tibetan language. QRNN consists of convolution components to extract features followed by an f-pooling component with a forget-gate to reduce the dimension of the features. The use of LSTM and CNN in the generator enables addressing the issue of processing longer sequences and parallel execution. The random initialisation of questions with Maximum Likelihood Estimation (MLE) ensures that both generated and original data follow a closer probability distribution.

Further optimisation proposes a reward strategy and Monte Carlo Search Strategy in the Reinforcement Learning model, which involves predicting the next sentence score based on the partially generated sequence rather than using the entire text. Following that, a BERT model facilitates the correction of the grammar of the generated text. The model evaluation uses data collected from the Tibetan website that involves 21783 questions for training different models with SeqGAN as the base model, QuGAN, QuGAN without Monte Carlo optimisation, QuGAN with BERT but without Monte Carlo Optimisation and QuGAN with BERT. QuGAN (Sun et al., 2019) has proven improvement of BLEU-2 score by 13.07 compared to the baseline with notable speed improvements. Further improvements can be made by generating grammatically correct questions by incorporating Tibetan grammar information and adding argument functions.

Another low-resource language scenario are the tasks involving regional dialects. A modified SentiGAN (Wang and Wan, 2018) based model (Carrasco et al., 2021) introduces an approach for data augmentation for Arabic Regional Dialects. Given that existing rich-annotated Dialectal Arabic datasets exhibit data scarcity, text data augmentation is also a solution for this issue. The selected regional Arabic dialects in that study are Egypt, Gulf, Maghreb, Levant, and Iraq. The generator uses an LSTM model with a policy gradient and a distractor using a CNN. Although the traditional SentiGAN (Wang and Wan, 2018) incorporates two sentiments, five dialects are generated using five generator/discriminator sets here. The model deviates from the other GAN-based text data augmentation models with a penalty instead of a reward for the discriminator model. The model generates a higher number of sentences than the original data size but with a reduced vocabulary size due to the usage of only the common words. The MADAR dataset is used for training and evaluating based on two new metrics to measure the novelty and diversity of the augmented texts and to assess further on four classification scenarios. Further improvement was also made by Wang and Wan (2018) by augmenting country-level dialects for Dialectical Arabic datasets.

In multilingual communities, loanwords are defined as words introduced and adopted from another language. Mi et al. (2021) provide data augmentation methodology to improve such loanword identification in low-resource language settings using a lexical-constrained GAN with two generators and a discriminator. It uses a log-linear RNN along with word and character-level embeddings, pronunciation similarity, and POS tagging features.

5.1.4 Fraud Detection

Social media platforms monitor user opinions on personal events, businesses, news, and politics. Market analysts use such reviews to come up with predictions and strategies to improve their business. To dominate the market, business owners may tend to add fake reviewers to their accounts or competitors' accounts. With the advancement of technology and bot usage, these fraud reviews are increasing exponentially. Hence, it is vital to identify such fraudulent reviews to perform a more reliable market analysis. There are different types of attempts in current research targeting fraud text detection, such as language models (Ott et al., 2011), behavioural profile analysis (Rayana and Akoglu, 2015) and deep learning feature representations (Le and Mikolov, 2014). A vital issue in fraud review identification is the lack of trusted labelled data, which leads to data scarcity of the models. To handle this problem, Aghakhani et al. (2018) proposed FakeGAN with one generator and two discriminators that address the model collapse problem, which is a typical problem for the GAN models. The training dataset X combines the subsets, X_T and X_D, which are fraud and real reviews, respectively. Z_g indicates all the reviews generated by FakeGAN. One discriminator, D, is defined for classifying fake ($X_D \cup Z_G$) and real X_T samples. Another discriminator, D′, is defined for classifying the generated samples similar to X_T and X_D. The model training follows the stochastic policy gradient method in reinforcement learning. Figure 5.5 illustrates an overview of FakeGAN, where the positive and negative samples are indicated by + and - symbols, respectively. The evaluation results of Aghakhani et al. (2018) indicate that the FakeGAN model performs similarly to the other fraud detection models in the literature. A main limitation of the model is the capability of generating reviews only in plain text without any association with the metadata, such as the rating scores. The possibility of bot-generated reviews in the training set as real samples and instability in the training process must also be addressed in future work. Further, another future research mentioned is the exploration of other GAN variants, such as Conditional GAN, and performing experiments with better hyperparameter tuning (Aghakhani et al., 2018).

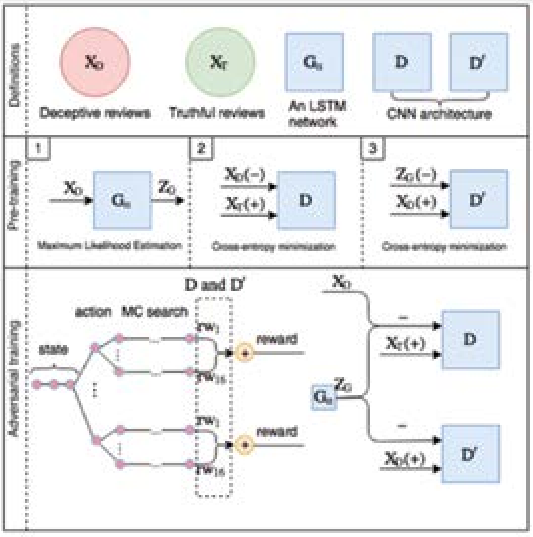


Figure 5.5: The overview of FakeGAN (Aghakhani et al., 2018)

The work proposed by Shehnepoor et al. (2022) addresses the drawbacks mentioned above of FakeGAN (Aghakhani et al., 2018) by generating score-correlated reviews using Information Gain Maximisation (IGM) theory to filter the fake samples that are generated. Their proposed model is called ScoreGAN, and it incorporates a given set of real reviews X, genuine reviews with scores $<X_g,S>$, fraud-human reviews with scores $<X_{fh},S>$ to generate score-correlated fraud bot reviews $<X_{fg},S>$. The overall fraud review set is $X_f = \{X_{fh},X_{fg}\}$. This model utilises two discriminators, D_g and D_f, following the FakeGAN architecture. The augmented data enables the discriminator D_g to distinguish bot-generated fraud reviews effectively. Figure 5.6 illustrates the framework of the ScoreGAN model. The information gain between the constraint c and the generator G_θ *(z,c)* is as follows:

$$I(c, G_\theta(z,c)) = H(c|G_\theta(z,c)) = -\mathbb{E}_{x\sim G_\theta(z,c),c\sim P(c|x)}[-\log P(c|x)] + H(c) \quad (14)$$

Using Lemma to address the issue of a fixed distribution on c, where H is the entropy definition, yields:

$$\begin{aligned} L(G_\theta, Q) &= -\mathbb{E}_{x\sim G_\theta(z,c),c\sim P(c|x)}[-\log Q(c|x)] + H(c) \\ &= \mathbb{E}_{x\sim G_\theta(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\ &\leq I(c, G_\theta(z,c)) \end{aligned} \quad (15)$$

The overall minimax game for is defined as follows:

$$max(\mathbb{E}_{x\sim X_g}[\log D_g(x)] + \mathbb{E}_{x\sim X_{fh}}[1 - \log D_g(x)] + \lambda L(G_\theta, Q)) \quad (16)$$

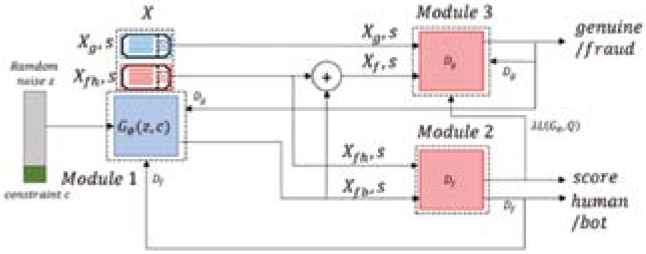


Figure 5.6: The illustration of the ScoreGAN model (Shehnepoor et al., 2022)

The evaluation results presented by Shehnepoor et al. (2022) showcase a 5% accuracy increase in Trip Advisor reviews and a 7% accuracy increase in Yelp reviews. Interestingly, experiments with a smaller subset of training data combined with augmented data are as effective as the full-sized datasets. A future direction in ScoreGAN would be to combine text features with other features, such as metadata (Shehnepoor et al., 2022).

Besides generating fraudulent reviews, social bots manipulate public opinions on different topics, accounts, and topics and spread malicious content. Due to the negative impacts that social bots impose, detecting and removing such fake accounts from social networks is nowadays crucial., It may lead to even more severe issues when the data generated by bots are more than those generated by genuine accounts because of the class imbalance issue. Wu et al. (2020) introduce an improved conditional GAN with a modified Gaussian Kernel Density Peak Clustering Algorithm (GKDPCA) to reduce noisy data generation and eliminate class imbalance within the data. The social bot detection framework uses a set of features: user-based, content and network. The use of Wasserstein distance with gradient penalty addresses the original conditional GAN model issues, which involve model collapse and the inability to control the category information in generated samples. As per the evaluation results, the improved cGAN outperforms three standard oversampling methods: random sampling (Liu et al., 2007), ADASYN (He et al., 2008) and SMOTE (Chawla et al., 2002) with a 97.56% of F1 score. As Wu et al. (2020) suggest, future work may head toward malicious bot detection incorporating other behavioural patterns and feature sequences.

Apart from the above applications, GAN text data augmentation has been employed for phishing URL detection to synthesise the training data (Xiao et al., 2021; Lee et al., 2020; Anand et al., 2018). Stanton and Irissappane (2019) present spamGAN for opinion spam detection that employs a semi-supervised GAN model.

5.1.5 Code-Switching Sentence Generation

Code-switching corresponds to the language changes in a given text. It may exist at the word or subword level when the editor writes different pieces in a text by changing it from one language to another. Chang et al. (2019) present an unsupervised GAN architecture to generate code-switching intra-sentences from monolingual data. Approaches to code-switching applications involve expensive human annotations and labelling speech data via transcription. (Chang et al., 2019) present a mechanism to generate such code-switching data without using any labelled data in the generator. Another application of GAN-based augmentation for code-switching is proposed by Gao et al. (2019) to generate intra-sentential code-switching sentences based on monolingual data, which outperforms code-switching language models. The future direction of Gao et al. (2019) will be towards enhancing the translator and generator.

5.1.6 Miscellaneous Applications

Large labelled dataset construction is a time-consuming process and requires domain expertise. Generative models with data augmentation are usually more sensitive to generating such categorically labelled data than complex manual annotation approaches. Most sentence generation models using GANs involve unlabelled texts, but it is also required to generate labelled data for a supervised classification task. There are two possible ways to perform this task: adding category information to the model or making the model generate a categorical sentence. The first approach loads the label information into the input representation. CS-GAN (Li et al., 2018) uses reinforcement learning, RNN and GAN-based category sentence generation to enlarge the original dataset. The sentiment analysis model by Li et al. (2018) performs well in supervised learning and shows the best performance with varying sentence lengths, even with smaller datasets with more categories.

Several other notable GAN application domains in text data augmentation include literary texts (Shahriar, 2022), multimodal news domain (Cadigan et al., 2021), controlled text generation (Betti et al., 2020; Malandrakis et al., 2019), machine translation (Ma et al., 2022; Fadaee et al., 2017; Sennrich et al., 2016) and medical domain (Kasthurirathne et al., 2021; Guan et al., 2018). These models either use GAN-synthesised data to mix with training data in pre-training or directly use the data generation alongside the training.

5.2 Critical Analysis of the Literature

Table 1 illustrates several applications of GAN text data augmentation in recent research in areas such as sentiment analysis, low resource language generation, fraud detection, code-switching sentence generation, and medical text generation, with a summary of approaches and future directions. Most models use SeqGAN architecture (Yu et al., 2017) with a few modifications in optimising the loss function. In category or label-based training, SentiGAN models Wang and Wan (2018) are adopted by providing label information and input features. Some of the models employ multiple generators or multiple discriminator architectures as well. Although not directly supporting text generation, Xiao et al. (2021) use Vanilla GAN to generate data GAN synthesised URLs. Future researchers could investigate enhancing these applications with a better combination of various features, enhancing training stability, extending to other languages, and building different GAN architectures.

Application	GAN Architecture	Approach	Suggested Future Directions
Sentiment Analysis	C-GAN (Gupta 2019)	Conditional GAN to augment data for sentiment classification with a generator, a discriminator, and a baseline classifier	Apply other GAN variants
	Seq-GAN (Luo et al., 2021)	Penalty-based SeqGAN to generate high-quality synthesised data	Use framework for other text domains
	G2S-AT-GAN (Chen et al., 2021)	Knowledge-graph-based rumour data augmentation (GERDA) and attention-based graph convolutions network with GAN	Address the problem of rumour data imbalance
	TransGAN (Shang et al., 2021)	RoBERTa model enhanced by a transformer-based GAN	Test the applicability of other datasets and cross-domain adaptation
Code-Switching Sentence Generation	Unsupervised GAN (Chang et al., 2019)	Unsupervised method to generate intra-sentential code-switching sentences using GAN	Improve translation accuracy
	CS-GAN (Gao et al., 2019)	Bert-C-based generator and discriminator	Generate a longer sequence of foreign words
Low-Resource Language Generation	QuGAN (Sun et al., 2019)	Tibetan question-answering corpus generation combining QuasiRNN and GAN	Increase the accuracy in generated corpus and add argument function and Tibetan grammar function
	Senti-GAN (Carrasco et al., 2021)	Sentimental GAN to generate sentences to overcome the data scarcity of the annotated Arabic regional dialects	Generate country-level dialects with data augmentation
	Lexical Controlled GAN (Mi et al., 2021)	Lexical constraint-based GAN to generate loanwords	Improve robustness of loanword identification with data augmentation

Table 1: Summary of GAN Text Augmentation Approaches

Application	GAN Architecture	Approach	Suggested Future Directions
Fraud Detection	Fake-GAN (Aghakhani et al., 2018)	Use two discriminator models and one generative model	Comparison with state-of-the-art supervised techniques
	Vanilla GAN (Xiao et al., 2021)	Use GAN-synthesised URLs to balance the datasets of legitimate and phishing URL	Explore the evolution pattern of the phishing websites
	Phish-GAN (Lee et al., 2020)	Use GAN to generate images of hieroglyphs conditioned on non-homoglyph input text images	Extend to other languages, such as Chinese and Korean
	C-GAN (Wu et al., 2020)	Improve the CGAN convergence issue by Wasserstein distance with a gradient penalty	Focus on malicious social bot detection
	Semi-Supervised GAN (Fadhel and Nyarko 2019)	Semi-supervised adversarial learning with discrete elements	Analysing the performance when incorporating the Movers distance measure
	Score-GAN (Shehnepoor et al., 2022)	Incorporate scores through IGM into the loss function	Combine text features with other behavioural features
Medical Text Generation	Seq-GAN (Kasthurirathne et al., 2021)	Generate synthetic free-text medical data with limited reidentification risk	
	mtGAN (Guan et al., 2018)	Generate synthetic texts of EMRs using reinforcement learning-based GAN	Explore hidden representations of medical texts

Table 1 (Continued): Summary of GAN Text Augmentation Approaches

6. Current Challenges and Future Research

The systematic review of GAN-based text data augmentation presented in this paper shows that many proposed frameworks for GAN-based text data augmentation still suffer from a lower accuracy for the classification tasks and the generation of grammatically incorrect long-textual data (Luo et al., 2021).

Evaluating the quality of the generated data is another potential gap in current research since there is a relatively lower number of attempts focusing on text data augmentation. There is still room for research on why and how data augmentation techniques provide accuracy improvements with a notion of in-depth theories and principles. In semantic classification methodologies involving data augmentation, it will be interesting to observe the impact of fake data generated on the opposition class via GANs to observe whether it will improve the model accuracy.

7. Conclusion

The paper provides a background study to showcase the recent research on GAN models as a text data augmentation tool. We used the PRISMA framework to ensure a non-biased and efficient paper search. With the notion of academic aspirations around data augmentation and GAN models, the paper presents a close view of applications spanning from sentence generation, addressing low resource languages, sentiment analysis and text analysis. Future directions in this area will further explore generating data distribution similar to but different from the original to reduce overfitting scenarios and new metrics to evaluate such text generation.

References

Aghakhani, H., Machiry, A., Nilizadeh, S., Krügel, C. and Vigna, G. (2018). Detecting Deceptive Reviews Using Generative Adversarial Networks. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE Computer Society, 89–95. Available at: https://doi.org/10.1109%2Fspw.2018.00022.

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N. and Zwerdling, N. (2020). Do Not Have Enough Data? Deep Learning to the Rescue!. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York: AAAI Press, 7383–7390. Available at: https://ojs.aaai.org/index. php/AAAI/article/view/6233.

Anand, A., Gorde, K., Antony Moniz, J. R., Park, N., Chakraborty, T. and Chu, B.-T. (2018). Phishing URL Detection with Oversampling based on Text Generative Adversarial Networks. In: Abe, N. et al., (eds.). *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 1168–1177. Available at: https:// doi.org/10.1109%2Fbigdata.2018.8622547.

Arjovsky, M. and Bottou, L. (2017). Towards Principled Methods for Training Generative Adversarial Networks. In: *5th International Conference on Learning Rep-*

resentations, ICLR 2017. OpenReview.net. Available at: https://openreview.net/forum?id=Hk4_qw5xe.

Betti, F., Ramponi, G. and Piccardi, M. (2020). Controlled Text Generation with Adversarial Learning. In: Davis, B. et al., (eds.). *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*. Association for Computational Linguistics, 29–34. Available at: https://aclanthology.org/2020.inlg-1.5/.

Cadigan, J., Sikka, K., Ye, M. and Graciarena, M. (2021). Resilient Data Augmentation Approaches to Multimodal Verification in the News Domain. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.*

Cao, R. and Lee, R. K.-W. (2020). HateGAN: Adversarial Generative-Based Data Augmentation for Hate Speech Detection. In: *Proceedings of the 28th International Conference on Computational Linguistics.* International Committee on Computational Linguistics, 6327–6338. Available at: https://aclanthology.org/2020.coling-main.557.

Carrasco, X. A., Elnagar, A. and Lataifeh, M. (2021). A Generative Adversarial Network for Data Augmentation: The Case of Arabic Regional Dialects. In: *Fifth International Conference On Arabic Computational Linguistics, ACLING 2021*. Online: Elsevier, 92–99. Available at: https://www.sciencedirect.com/science/article/pii/S1877050921011674.

Chang, C.-T., Chuang, S.-P. and Lee, H. (2019). Code-switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. In: Kubin, G. and Kacic, Z., (eds.). *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, 554–558. Available at: https://doi.org/10.21437%2Finterspeech.2019-3214.

Chawla, N. v, Bowyer, K. W., Hall, L. O. and Kegelmeyer, W.P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. Available at: https://doi.org/10.1613%2Fjair.953.

Chen, H., Ji, Y. and Evans, D. (2020). Finding Friends and Flipping Frenemies: Automatic Paraphrase Dataset Augmentation Using Graph Theory. In: Cohn, T., He, Y., and Liu, Y., (eds.). *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, 4741–4751. Available at: https://aclanthology.org/2020.findings-emnlp.426.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. and Abbeel, P. (2016). Infogan: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In: Lee, D. D. et al., (eds.). *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook: Curran Associates Inc., 2172–2180. Available at: https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html.

Chen, X., Zhu, D., Lin, D. and Cao, D. (2021). Rumor Knowledge Embedding Based Data Augmentation for Imbalanced Rumor Detection. *Information Sciences*, 580, 352–370. Available at: https://doi.org/10.1016/j.ins.2021.08.059.

Donahue, D. and Rumshisky, A. (2018). Adversarial Text Generation Without Reinforcement Learning. *CoRR*, abs/1810.06640. Available at: http://arxiv.org/abs/1810.06640.

Fabbri, A., Han, S., Li, H., Li, H., Ghazvininejad, M., Joty, S., Radev, D. and Mehdad, Y. (2021). Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation. In: Toutanova, K. et al., (eds.). *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 704–717. Available at: https://aclanthology.org/2021.naacl-main.57.

Fadaee, M., Bisazza, A. and Monz, C. (2017). Data Augmentation for Low-Resource Neural Machine Translation. In: Barzilay, R. and Kan, M.-Y., (eds.). *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 567–573. Available at: https://aclanthology.org/P17-2090.

Fadhel, M. ben and Nyarko, K. (2019). GAN Augmented Text Anomaly Detection with Sequences of Deep Statistics. In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 1–5. Available at: https://doi.org/10.1109/CISS.2019.8693024.

Fedus, W., Goodfellow, I. J. and Dai, A. M. (2018). MaskGAN: Better Text Generation via Filling in the ______. In: *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*. Available at: https://openreview.net/pdf?id=ByOExmWAb.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T. and Hovy, E. (2021). A Survey of Data Augmentation Approaches for NLP. In: Zong, C. et al., (eds.). *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 968–988. Available at: https://aclanthology.org/2021.findings-acl.84.

Feng, S. Y., Gangal, V., Kang, D., Mitamura, T. and Hovy, E. (2020). GenAug: Data Augmentation for Finetuning Text Generators. In: *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Online: Association for Computational Linguistics, 29–42. Available at: https://aclanthology.org/2020.deelio-1.4.

Gao, Y., Feng, J., Liu, Y., Hou, L., Pan, X. and Ma, Y. (2019). Code-Switching Sentence Generation by Bert and Generative Adversarial Networks. In: Kubin, G. and Kacic, Z., (eds.). *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, 3525–3529. Available at: https://doi.org/10.21437%2Finterspeech.2019-2501.

Goldberg, A. and Zhu, X. (2006). Seeing Stars When There Aren't Many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization. In: *Proceedings of TextGraphs: The First Workshop on Graph Based Methods for Natural Language Processing*. Association for Computational Linguistics, 45–52. Available at: https://aclanthology.org/W06-3808.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C. and Bengio, Y. (2014). Generative Adversarial Nets. In: Ghahramani, Z. et al., (eds.). *Advances in Neural Information Processing Systems 27: Annual Conference on NIPS 2014*, 2672–2680. Available at: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Guan, J., Li, R., Yu, S. and Zhang, X. (2018). Generation of Synthetic Electronic Medical Record Text. In: Zheng, H. J. et al., (eds.). *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE Computer Society, 374–380. Available at: https://doi.org/10.1109%2Fbibm.2018.8621223.

Guo, H. (2020). Nonlinear Mixup: Out-Of-Manifold Data Augmentation for Text Classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, 4044–4051. Available at: https://ojs.aaai.org/index.php/AAAI/article/view/5822.

Gupta, R. (2019). Data Augmentation for Low Resource Sentiment Analysis Using Generative Adversarial Networks. In: *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7380–7384. Available at: https://doi.org/10.1109%2Ficassp.2019.8682544.

Gupta, R., Sahu, S., Espy-Wilson, C. Y. and Narayanan, S. S. (2018). Semi-Supervised and Transfer Learning Approaches for Low Resource Sentiment Classification. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5109–5113. Available at: https://doi.org/10.1109/ICASSP.2018.8461414.

He, H., Bai, Y., Garcia, E. A. and Li, S. (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 1322–1328. Available at: https://doi.org/10.1109/IJCNN.2008.4633969.

Hou, Y., Liu, Y., Che, W. and Liu, T. (2018). Sequence-to-Sequence Data Augmentation for Dialogue Language Understanding. In: Bender, E. M., Derczynski, L., and Isabelle, P., (eds.). *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*. Association for Computational Linguistics, 1234–1245. Available at: https://aclanthology.org/C18-1105.

Jia, R., Raghunathan, A., Göksel, K. and Liang, P. (2019). Certified Robustness to Adversarial Word Substitutions. In: Inui, K. et al., (eds.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 4129–4142. Available at: https://aclanthology.org/D19-1423.

Kang, D., Khot, T., Sabharwal, A. and Hovy, E. (2018). AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples. In: Gurevych, I. and Miyao, Y., (eds.). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2418–2428. Available at: https://aclanthology.org/P18-1225.

Karras, T., Laine, S. and Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation/IEEE, 4396–4405. Available at: https://doi.org/10.1109%2Fcvpr.2019.00453.

Kasthurirathne, S. N., Dexter, G. and Grannis, S. (2021). Generative Adversarial Networks for Creating Synthetic Free-Text Medical Data: A Proposal for Collaborative Research and Re-use of Machine Learning Models. In: *Proceedings – AMIA Joint Summits Translational Science*, 335–344.

Kim, H.-Y., Roh, Y.-H. and Kim, Y.-K. (2019). Data Augmentation by Data Noising for Open-vocabulary Slots in Spoken Language Understanding. In: Kar, S. et al., (eds.). *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, 97–102. Available at: https://aclanthology.org/N19-3014.

Kumar, A., Bhattamishra, S., Bhandari, M. and Talukdar, P. (2019). Submodular Optimization-based Diverse Paraphrasing and its Effectiveness in Data Augmentation. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 3609–3619. Available at: https://aclanthology.org/N19-1363.

Kusner, M. J. and Hernández-Lobato, J. M. (2016). GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *CoRR*, abs/1611.04051. Available at: http://arxiv.org/abs/1611.04051.

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning – Volume 32*. PMLR, 1188–1196. Available at: http://proceedings.mlr.press/v32/le14.html.

Lee, J. S., Yam, G. P. D. and Chan, J. H. (2020). PhishGAN: Data Augmentation and Identification of Homoglpyh Attacks. In: *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. IEEE, 1–6. Available at: https://doi.org/10.1109%2Fccci49893.2020.9256804.

Li, Y., Pan, Q., Wang, S., Yang, T. and Cambria, E. (2018). A Generative Model for Category Text Generation. *Information Sciences*, 450, 301–315. Available at: https://doi.org/10.1016%2Fj.ins.2018.03.050.

Liu, A. Y., Ghosh, J. and Martin, C. E. (2007). Generative Oversampling for Mining Imbalanced Datasets. In: Stahlbock, R., Crone, S. F., and Lessmann, S., (eds.).

Proceedings of the 2007 International Conference on Data Mining, DMIN. CSREA Press, 66–72.

Liu, M.-Y. and Tuzel, O. (2016). Coupled Generative Adversarial Networks. In: Lee, D. D. et al., (eds.). *Advances in Neural Information Processing Systems 29: Annual Conference on NIPS 2016*. NeurIPS, 469–477. Available at: https://proceedings.neurips.cc/paper/2016/hash/502e4a16930e414107ee22b6198c578f-Abstract.html.

Longpre, S., Lu, Y., Tu, Z. and DuBois, C. (2019). An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering. In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics, 220–227. Available at: https://aclanthology.org/D19-5829.

Louvan, S. and Magnini, B. (2020). Simple is Better! Lightweight Data Augmentation for Low Resource Slot Filling and Intent Classification. In: Nguyen, M. le, Luong, M. C., and Song, S., (eds.). *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*. Association for Computational Linguistics, 167–177. Available at: https://aclanthology.org/2020.paclic-1.20.

Lu, K., Mardziel, P., Wu, F., Amancharla, P. and Datta, A. (2020). Gender Bias in Neural Natural Language Processing. In: Nigam, V. et al., (eds.). *Logic, Language, and Security – Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*. Springer, 189–202. Available at: https://doi.org/10.1007/978-3-030-62077-6_14.

Luo, J., Bouazizi, M. and Ohtsuki, T. (2021). Data Augmentation for Sentiment Analysis Using Sentence Compression-Based SeqGAN With Data Screening. *IEEE*, 9, 99922–99931. Available at: https://doi.org/10.1109%2Faccess.2021.3094023.

Ma, W., Yan, B. and Sun, L. (2022). Generative Adversarial Network-based Short Sequence Machine Translation from Chinese to English. *Scientific Programming*, 2022, 1–10. Available at: https://doi.org/10.1155%2F2022%2F7700467.

Malandrakis, N., Shen, M., Goyal, A., Gao, S., Sethi, A. and Metallinou, A. (2019). Controlled Text Generation for Data Augmentation in Intelligent Artificial Agents. In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Association for Computational Linguistics, 90–98. Available at: https://aclanthology.org/D19-5609.

Mao, X., Wang, Y., Liu, X. and Guo, Y. (2017). An Adaptive Weighted Least Square Support Vector Regression for Hysteresis in Piezoelectric Actuators. *Sensors and Actuators A: Physical*, 263, 423–429. Available at: https://doi.org/10.1016%2Fj.sna.2017.06.030.

Mi, C., Zhu, S. and Nie, R. (2021). Improving Loanword Identification in Low-Resource Language with Data Augmentation and Multiple Feature Fusion. *Computational Intelligence and Neuroscience*, 2021, 1–9. Available at: https://doi.org/10.1155%2F2021%2F9975078.

Mimura, M. (2020). Using Fake Text Vectors to Improve the Sensitivity of Minority Class for Macro Malware Detection. *Journal of Information Security and Ap-*

plications, 54, 102600. Available at: https://www.sciencedirect.com/science/article/pii/S2214212620307651.

Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784. Available at: http://arxiv.org/abs/1411.1784.

Moher, D., Liberati, A., Tetzlaff, J. and Altman, D. G. (2009). Preferred Reporting Items for Systematic Reviews and Meta-analyses: the PRISMA statement. *BMJ*, 339, b2535–b2535. Available at: https://www.bmj.com/content/339/bmj.b2535.

Nowozin, S., Cseke, B. and Tomioka, R. (2016). f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In: Lee, D. D. et al., (eds.). *Advances in Neural Information Processing Systems 29: Annual Conference on NIPS 2016*. NeurIPS, 271–279. Available at: https://proceedings.neurips.cc/paper/2016/hash/cedebb6e872f539bef8c3f919874e9d7-Abstract.html.

Ott, M., Choi, Y., Cardie, C. and Hancock, J. T. (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In: Lin, D., Matsumoto, Y., and Mihalcea, R., (eds.). *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 309–319. Available at: https://aclanthology.org/P11-1032.

Parida, S. and Motlicek, P. (2019). Abstract Text Summarization: A Low Resource Challenge. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 5994–5998. Available at: https://aclanthology.org/D19-1616.

Park, T., Liu, M.-Y., Wang, T.-C. and Zhu, J.-Y. (2019). GauGAN: Semantic Image Synthesis with Spatially Adaptive Normalization. In: *ACM SIGGRAPH 2019 Real-Time Live!* Association for Computing Machinery. Available at: https://doi.org/10.1145%2F3306305.3332370.

Paschali, M., Simson, W., Roy, A. G., Naeem, M.F., Göbl, R., Wachinger, C. and Navab, N. (2019). Manifold Exploring Data Augmentation with Geometric Transformations for Increased Performance and Robustness. In: Chung Albert C. S. and Gee, J. C. and Y. P. A. and B. S., (eds.). *Information Processing in Medical Imaging*. Cham: Springer International Publishing, 517–529. Available at: https://doi.org/10.1007%2F978-3-030-20351-1_40.

Qin, L., Ni, M., Zhang, Y. and Che, W. (2021). CoSDA-ML: Multi-Lingual Code-Switching Data Augmentation for Zero-Shot Cross-Lingual NLP. In: Bessiere, C., (ed.). *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. IJCAI'20. International Joint Conferences on Artificial Intelligence Organization, 3853–3860. Available at: https://doi.org/10.24963/ijcai.2020/533.

Quan, J. and Xiong, D. (2019). Effective Data Augmentation Approaches to End-to-End Task-Oriented Dialogue. In: 2*019 International Conference on Asian Language Processing (IALP)*. IEEE, 47–52. Available at: https://doi.org/10.1109%-2Fialp48816.2019.9037690.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8), 9.

Radford, A., Metz, L. and Chintala, S. (2018). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: *2018 37th Chinese Control Conference (CCC)*. IEEE, 9159–9163. Available at: https://doi.org/10.23919%2Fchicc.2018.8482813.

Rayana, S. and Akoglu, L. (2015). Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Association for Computing Machinery, 985–994. Available at: https://doi.org/10.1145/2783258.2783370.

Riabi, A., Scialom, T., Keraron, R., Sagot, B., Seddah, D. and Staiano, J. (2021). Synthetic Data Augmentation for Zero-Shot Cross-Lingual Question Answering. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 7016–7030. Available at: https://aclanthology.org/2021.emnlp-main.562.

Şahin, G.G. and Steedman, M. (2018). Data Augmentation via Dependency Tree Morphing for Low-Resource Languages. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 5004–5009. Available at: https://aclanthology.org/D18-1545.

Sennrich, R., Haddow, B. and Birch, A. (2016). Improving Neural Machine Translation Models with Monolingual Data. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin: Association for Computational Linguistics, 86–96. Available at: https://aclanthology.org/P16-1009.

Shahriar, S. (2022). GAN Computers Generate Arts? A Survey on Visual Arts, Music, and Literary Text Generation using Generative Adversarial Network. *Displays*, 73, 102237. Available at: https://www.sciencedirect.com/science/article/pii/S0141938222000658.

Shang, Y., Su, X., Xiao, Z. and Chen, Z. (2021). Campus Sentiment Analysis with GAN-based Data Augmentation. In: *13th International Conference on Advanced Infocomm Technology (ICAIT)*. IEEE, 209–214. Available at: https://doi.org/10.1109%2Ficait52638.2021.9702068.

Shehnepoor, S., Togneri, R., Liu, W. and Bennamoun, M. (2022). ScoreGAN: A Fraud Review Detector Based on Regulated GAN With Data Augmentation. *IEEE Transactions on Information Forensics and Security*, 17, 280–291.

Sindhwani, V. and Melville, P. (2008). Document-Word Co-regularization for Semi-supervised Sentiment Analysis. In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE Computer Society, 1025–1030. Available at: https://doi.org/10.1109/ICDM.2008.113.

Socher, R., Pennington, J., Huang, E. H.-C., Ng, A. and Manning, C. D. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions.

In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 151–161. Available at: https://aclanthology.org/D11-1014/.

Stanton, G. and Irissappane, A. A. (2019). GANs for Semi-Supervised Opinion Spam Detection. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 5204–5210. Available at: https://doi.org/10.24963/ijcai.2019/723.

Sun, Y., Chen, C., Xia, T. and Zhao, X. (2019). QuGAN: Quasi Generative Adversarial Network for Tibetan Question Answering Corpus Generation. *IEEE Access*, 7, 116247–116255. Available at: https://doi.org/10.1109%2Faccess.2019.2934581.

Täckström, O. and McDonald, R. T. (2011). Semi-supervised Latent Variable Models for Sentence-level Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 569–574. Available at: https://aclanthology.org/P11-2100.

Wang, K. and Wan, X. (2018). SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 4446–4452. Available at: https://doi.org/10.24963%2Fijcai.2018%2F618.

Wei, J. and Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 6382–6388. Available at: https://aclanthology.org/D19-1670.

Wu, B., Liu, L., Yang, Y., Zheng, K. and Wang, X. (2020). Using Improved Conditional Generative Adversarial Networks to Detect Social Bots on Twitter. *IEEE Access*, 8, 36664–36680. Available at: https://doi.org/10.1109%2Faccess.2020.2975630.

Wu, J., Zhang, C., Xue, T., Freeman, B. and Tenenbaum, J. (2016). Learning a Probabilistic Latent Space of Object Shapes via 3d Generative-adversarial Modeling. In: Lee, D. D. et al., (eds.). *Advances in neural information processing systems*. Barcelona, 82–90. Available at: https://proceedings.neurips.cc/paper/2016/hash/44f683a84163b3523afe57c2e008bc8c-Abstract.html.

Xiao, X., Xiao, W., Zhang, D., Zhang, B., Hu, G., Li, Q. and Xia, S. (2021). Phishing Websites Detection via CNN and Multi-head Self-attention on Imbalanced Datasets. *Computers & Security*, 108, 102372. Available at: https://www.sciencedirect.com/science/article/pii/S0167404821001966.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T. and Le, Q. v. (2020). Unsupervised Data Augmentation for Consistency Training. In: Larochelle, H. et al., (eds.). *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Red Hook: Curran Associates Inc., 6256–6268. Available at: https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html.

Yang, W., Xie, Y., Tan, L., Xiong, K., Li, M. and Lin, J. J. (2019). Data Augmentation for BERT Fine-Tuning in Open-Domain Question Answering. *CoRR*, abs/1904.06652. Available at: https://arxiv.org/abs/1904.06652.

Yang, Y., Malaviya, C., Fernandez, J., Swayamdipta, S., le Bras, R., Wang, J.-P., Bhagavatula, C., Choi, Y. and Downey, D. (2020). Generative Data Augmentation for Commonsense Reasoning. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, 1008–1025. Available at: https://aclanthology.org/2020.findings-emnlp.90.

Yu, L., Zhang, W., Wang, J. and Yu, Y. (2017). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In: Singh, S. and Markovitch, S., (eds.). *AAAI Conference on Artificial Intelligence*. AAAI Press, 2852–2858. Available at: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J. and Yoo, Y. J. (2019). CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 6022–6031. Available at: https://doi.org/10.1109%2Ficcv.2019.00612.

Zhang, H., Cissé, M., Dauphin, Y. and Lopez-Paz, D. (2018). Mixup: Beyond Empirical Risk Minimization. In: *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net. Available at: https://openreview.net/forum?id=r1Ddp1-Rb.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. and Metaxas, D.N. (2017). StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 5908–5916. Available at: https://doi.org/10.1109%2Ficcv.2017.629.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V. and Chang, K.-W. (2018). Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 15–20. Available at: https://aclanthology.org/N18-2003.

Zhu, H., Dong, L., Wei, F., Qin, B. and Liu, T. (2022). Transforming Wikipedia into Augmented Data for Query-Focused Summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2357–2367. Available at: https://doi.org/10.1109%2Ftaslp.2022.3171963.

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2242–2251. Available at: https://doi.org/10.1109%2Ficcv.2017.244.

https://doi.org/10.33919/JCAL.23.1.2

EXTRACTING ALGORITHMIC COMPLEXITY IN SCIENTIFIC LITERATURE FOR ADVANCE SEARCHING

Abu Bakar[1], Raheem Sarwar[2, *], Saeed-Ul Hassan[3],
Raheel Nawaz[4]

[1]Computer Science, Information Technology University, Lahore, Pakistan
[2] Department of Operations, Technology, Events and Hospitality Management, Manchester Metropolitan University, United Kingdom
[3] Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom
[4] Staffordshire University, United Kingdom
[*]Corresponding Author (R.Sarwar@mmu.ac.uk)

Abstract

Non-textual document elements such as charts, diagrams, algorithms and tables play an important role to present key information in scientific documents. Recent advances in information retrieval systems tap this information to answer more complex user queries by mining text pertaining to non-textual document elements from full text. Algorithms are critically important in computer science. Researchers are working on existing algorithms to improve them for critical application. Moreover, new algorithms for unsolved and newly faced problems are under development. These enhanced and new algorithms are mostly published in scholarly documents. The complexity of these algorithms is also discussed in the same document by the authors. Complexity of an algorithm is also an important factor for information retrieval (IR) systems. In this paper, we mine the relevant complexities of algorithms from full text document by comparing the metadata of the algorithm, such as caption and function name, with the context of the paragraph in which complexity related discussion is made by the authors. Using the dataset of 256 documents downloaded from CiteSeerX repository, we manually annotate 417 links between algorithms and their complexities. Further, we apply our novel rule-based approach that identifies the desired links with 81% precision, 75% recall, 78% F1-score and 65% accuracy. Overall, our method of identifying the links has potential to improve information retrieval systems that tap the advancements of full text and more specifically non-textual document elements.

Keywords: *algorithm search, algorithm complexity, information retrieval, non-textual document elements (NTDE)*

1. Introduction

Academic literature is growing exponentially in last two decades and flourishing in an unprecedented pace, which has brought new challenges to information retrieval research (Khan, Liu, Shakil and Alam, 2017). Non-Textual Document Elements (NTDEs) such as figures, charts, pseudo-codes, and tables are very common in scientific documents, and they are vital elements for communicating the key information. These elements are sometime placed at the start or at the end of the page instead of following the flow of document text, and the discussion about these elements may or may not be on the same page, the discussion about these elements mostly has the reference of the caption of the entity. Sometimes, these elements are referred multiple times in different sections of the scholarly document.

Algorithms are well-defined methodologies to solve the problems and they are important in every field of science and technology. There are many features of an algorithm such as correctness, elegance, efficiency and scalability. Efficiency of an algorithm is defined as complexity of algorithm, and it is based on time and space. The main aim of estimating the complexity of algorithms is to categorize them according to their efficiency. Complexity of algorithm is described asymptotically using three types of notations as:

(1) O-notation (big oh notation) used for upper bound, (2) Ω-notation (big omega notation) used for lower bound and (3) Θ-notation (big theta notation) used for tight bound. We have used these asymptotic notations to identify the complexity lines in full text scholarly documents.

Scholarly publications host a tremendous number of high-quality algorithms, developed by professionals and researchers. Normally, when new algorithms are published, or existing algorithms are enhanced, their time and/or space complexities are also discussed in the same document by the authors. Most of the time authors are not working on algorithms or not trying to improve existing ones, they are publishing the algorithms that they have used in their research, and sometimes they do not discuss the complexity of the algorithm because it is well-known, or because they do not focus on it.

The complexities of an algorithm (for time and space) can be identified from the document by analyzing the context of the paragraph in which the complexity is mentioned, and the metadata (such as the algorithm caption and the algorithm label) of the algorithm extracted from the same document.

Our Research Contributions. In this research, our contributions are as follows:

- Identification of algorithmic complexity lines in full text document using regular expressions and synopsis generation for each complexity line.
- Algorithmic metadata compilation of algorithms – normally, there are multiple algorithms in a scholarly document and the metadata of each algorithm are compiled separately.
- Linking complexity related textual lines to algorithmic metadata using a novel rule-based approach.

In Section 2, we have discussed related work. In Section 3, a dataset is described, and a system model of our research is discussed in Section 4. Experiments and their results are given in Section 5 and Section 6 concludes the research with future suggestions.

2. Literature Review and Related Work

Many algorithms are being published in research articles on a monthly and yearly basis (Bhatia, Tuarob, Mitra and Giles, 2011). Hundreds of articles are published and/or added to digital archives on a monthly and yearly basis and the arXiv[1] has crossed the boundary of 1.3M full text publications, which shows the importance of this research.

[1] https://arxiv.org/stats/monthly_submissions.

2.1 Algorithmic Representations

Normally, algorithms are explained in pseudo-code (PC) (as shown in Figure 1), in natural language as algorithmic-procedure (AP) (as shown in Figure 2), in mathematical formatting (as shown in Figure 3) or in coding style (as shown in Figure 4) (Tuarob, Bhatia, Mitra and Giles, 2016). Algorithms can be implemented in any programming language. There are a number of document elements in a scholarly document such as figures (Siegel, Horvitz, Levin, Divvala and Farhadi, 2016), tables (Liu, Bai, Mitra and Giles, 2007), forms (Coüasnon and Lemaitre, 2014), algorithms (Bhatia, Mitra and Giles, 2010) mathematical expressions (Baker, Sexton, Sorge and Suzuki, 2011) (Zanibbi and Blostein, 2012), programing codes and a number of textual sections such as abstract, acknowledgments (Khabsa, Treeratpituk and Giles, 2012), collaborations (Chen, Gou, Zhang and Giles, 2011), methodology, results (e.g., precision, recall, or F-measure), conclusion and references. Algorithms are normally given as figures (Bhatia and Mitra, 2012), in mathematical formatting, in coding style, or sparse boxes like other document elements (Tuarob, Bhatia, Mitra and Giles, 2013).

2.2 Information Retrieval Systems and Algorithms

Information retrieval (IR) is a technique for searching required or relevant information form exiting data (Wang, 2009). There are several search engines to search for academic literature such as Google Scholar[2], Microsoft Academic[3], PloS One[4], Semantic Scholar[5], Science Direct[6], ACM Digital Library[7] and CiteSeerX[8], an academic document search (Wu, et al., 2015). There are also some search engines that are optimized for a specific area of science such as BioText Search Engine[9] which is optimized for bioinformatics related search (Hearst, et al., 2007). There are some other IR systems such as TableSeer for searching tables in digital libraries (Liu, Bai, Mitra and

[2] https://scholar.google.com/.

[3] https://academic.microsoft.com/.

[4] http://journals.plos.org/plosone/.

[5] https://www.semanticscholar.org.

[6] https://www.sciencedirect.com/.

[7] https://dl.acm.org.

[8] http://citeseerx.ist.psu.edu/index.

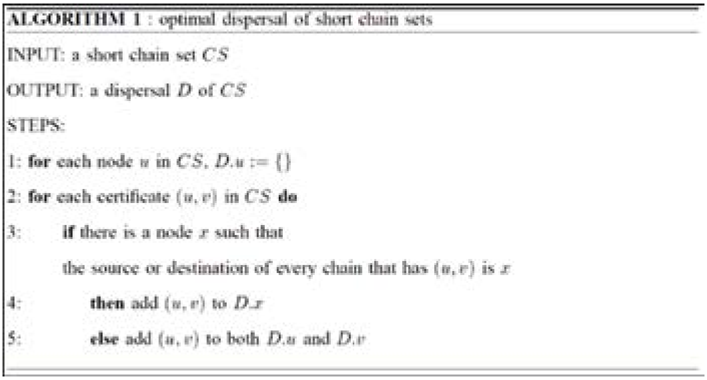
[9] http://biosearch.berkeley.edu/

Giles, 2007), AckSeer for acknowledgments (Khabsa, Treeratpituk and Giles, 2012), CollabSeer for collaborations (Chen, Gou, Zhang and Giles, 2011), FigureSeer for figures (Siegel, Horvitz, Levin, Divvala and Farhadi, 2016) and AlgorithmSeer for searching algorithms in scholarly big data (Tuarob, Bhatia, Mitra and Giles, 2016).

Efficient algorithms are critically important and sometime crucial for certain software projects. Nowadays, there are a number of source code search engines for software developers and researchers to find relevant source code according to their requirements. Information retrieval systems for algorithms in scholarly documents are improving in past recent years to fulfill the search queries by providing relevant information such as Sourerer (Bajracharya, Ossher and Lopes, 2009) and Exemplar (EXEcutable exaMPLes ARchive) (McMillan, Grechanik, Poshyvanyk, Fu and Xie, 2012).

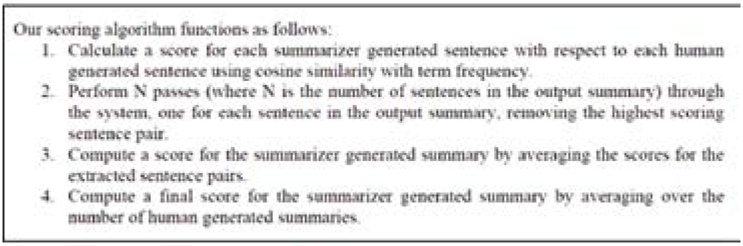
In recent years, a few attempts have been made to extract non textual document elements such as figures, tables, algorithms and charts (Al-Zaidy and Giles, 2017), (Tuarob, Bhatia, Mitra and Giles, 2013), (Safder, Hassan and Aljohani, 2018). These techniques are actively applied for effective document summarization to improve the existing IR systems. A customized search engine AlgorithmSeer is designed for algorithms searching from full text articles (Tuarob, Bhatia, Mitra and Giles, 2016). This system uses some rule-based and machine learning based techniques for automatic extraction of algorithms from full text articles, then it creates a specialized algorithmic summary of a document to match against a user search query.

Moreover, to mine information from results figures present in scholarly articles, FigureSeer, a specialized results figure extractor system has been presented (Siegel, Horvitz, Levin, Divvala and Farhadi, 2016). The designed system leveraged the deep learning-based techniques to identify, class an image as results image. Further, the system mines the information presented on these figures to design a results figure search engine. Likewise, another system Deep-Figures has implemented a similar kind of system using supervised neural network-based technique (Siegel, Lourie, Power and Ammar, 2018). Table search system is also a very prominent work to retrieve complex tables against user queries from massive repositories (Nargesian, Zhu, Pu and Miller, 2018). Additionally, linking these document elements (table, figure, algorithms) with their discussions or reference sentences written in the full body text of an article has its own significance. This additional information about a document element can help to understand its context instead of reading and scrolling the whole paper (Bhatia and Mitra, 2012).



```
ALGORITHM 1 : optimal dispersal of short chain sets
INPUT: a short chain set $CS$
OUTPUT: a dispersal $D$ of $CS$
STEPS:
for each node $u$ in $CS$, $D.u := \{\}$
for each certificate $(u, v)$ in $CS$ do
    if there is a node $x$ such that
    the source or destination of every chain that has $(u, v)$ is $x$
        then add $(u, v)$ to $D.x$
        else add $(u, v)$ to both $D.u$ and $D.v$
```

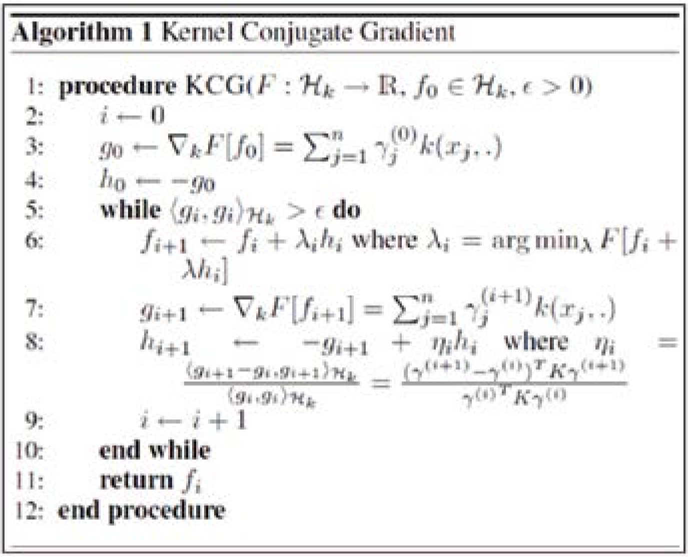
Figure 1: Example of Pseudo-Code (PC) **(Jung, Elmallah and Gouda, 2007)**



Our scoring algorithm functions as follows:

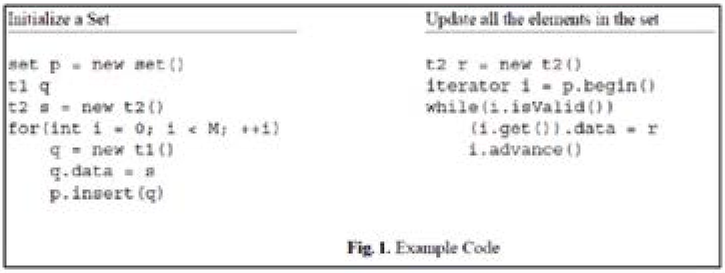
1. Calculate a score for each summarizer generated sentence with respect to each human generated sentence using cosine similarity with term frequency.
2. Perform N passes (where N is the number of sentences in the output summary) through the system, one for each sentence in the output summary, removing the highest scoring sentence pair.
3. Compute a score for the summarizer generated summary by averaging the scores for the extracted sentence pairs.
4. Compute a final score for the summarizer generated summary by averaging over the number of human generated summaries.

Figure 2: Example of Algorithmic Procedure (AP), from **(Stewart and Callan, 2009)**



```
Algorithm 1 Kernel Conjugate Gradient
procedure KCG($F : \mathcal{H}_k \to \mathbb{R}, f_0 \in \mathcal{H}_k, \epsilon > 0$)
    $i \leftarrow 0$
    $g_0 \leftarrow \nabla_k F[f_0] = \sum_{j=1}^{n} \gamma_j^{(0)} k(x_j, .)$
    $h_0 \leftarrow -g_0$
    while $\langle g_i, g_i \rangle_{\mathcal{H}_k} > \epsilon$ do
        $f_{i+1} \leftarrow f_i + \lambda_i h_i$ where $\lambda_i = \arg\min_\lambda F[f_i + \lambda h_i]$
        $g_{i+1} \leftarrow \nabla_k F[f_{i+1}] = \sum_{j=1}^{n} \gamma_j^{(i+1)} k(x_j, .)$
        $h_{i+1} \leftarrow -g_{i+1} + \eta_i h_i$ where $\eta_i = \frac{\langle g_{i+1} - g_i, g_{i+1} \rangle_{\mathcal{H}_k}}{\langle g_i, g_i \rangle_{\mathcal{H}_k}} = \frac{(\gamma^{(i+1)} - \gamma^{(i)})^T K \gamma^{(i+1)}}{\gamma^{(i)T} K \gamma^{(i)}}$
        $i \leftarrow i + 1$
    end while
    return $f_i$
end procedure
```

Figure 3: Example of Mathematical Formatting **(Ratliff and Bagnell, 2007)**



Initialize a Set

```
set p = new set()
t1 q
t2 s = new t2()
for(int i = 0; i < M; ++i)
    q = new t1()
    q.data = s
    p.insert(q)
```

Update all the elements in the set

```
t2 r = new t2()
iterator i = p.begin()
while(i.isValid())
    (i.get()).data = r
    i.advance()
```

Fig. 1. Example Code

Figure 4: Example of Coding Style Algorithm
(Marron, Stefanovic, Hermenegildo and Kapur, 2007)

Generally, run time complexity related to an algorithm is mentioned in the full body text of a document as algorithmic metadata. In order to find out the above-mentioned complexities of an algorithm, we need to link the algorithm and the paragraphs in which the complexity of that algorithm is discussed. Recently, a few techniques have been designed to extract evolution results lines related to an algorithm from full text articles (Safder, Sarfraz, Hassan, Ali and Tuarob, 2017). However, to the best of our knowledge no work has been done to find run time complexities and to link these run time complexities with their respective algorithm in full text scholarly publications.

2.3 Algorithm Detection in Scholarly Documents

A number of rule-based, machine learning-based, and deep learning-based methods have been designed for detection and extraction of algorithms from full-text scholarly documents (Tuarob, Bhatia, Mitra and Giles, 2016), (Safder, Sarfraz, Hassan, Ali and Tuarob, 2017), (Safder, Hassan and Aljohani, 2018), (Lai, Xu, Liu and Zhao, 2015). Detection of an algorithm in a document is the first step, the aim is to make it retrievable on users' queries. For IR system algorithms metadata are needed, therefore caption lines, indication sentences, function names or algorithm labels are extracted from documents related to the algorithms for metadata.

Results related to algorithmic evaluation performance such as precision, recall, F-measure and accuracy are also extracted from the same documents to improve results for developers and researchers. Complexity of algorithm is also an important factor for IR systems; currently it not directly used in IR systems as effectively as it is important. We aim to improve the feature of time and space complexity identification from scholarly documents in our research. In this paper, we designed a mechanism to identify complexity lines and then to link these complexity lines with their relevant algorithm.

3. Data and Limitations

There are over 100 million scholarly documents in the English language on the web in different fields (Khabsa and Giles, 2014), most of which are indexed by digital libraries. For our research we have selected a small dataset of 258 documents, as discussed below.

Data. The dataset is selected from (Safder, Sarfraz, Hassan, Ali and Tuarob, 2017), which consists of 258 documents originally selected from the CiteSeerX repository (Tuarob, Bhatia, Mitra and Giles, 2016), and has 37,000 lines of text. The data is manually labeled: 2,331 lines for algorithmic efficiency and 80 lines for algorithmic time complexity. There are some limitations to using this dataset which are discussed in Section 3.2. We use algorithmic metadata lines tagging as given in the section below[10]. Data is also tagged for the following type of lines related to algorithmic metadata (as shown in Figure 5): pseudo-code lines, pseudo-code caption lines, function name, algorithm label, indication sentence, algorithm section header, explanation sentence and proposal sentence. We use tagging to identify algorithms and their metadata, and then we compare this metadata with the paragraph in which the complexity is mentioned.

There are 142 documents in the dataset, in which we found algorithms and other tagged lines related to algorithms (tagging is listed above). There are 62 documents in which we found complexity lines, and only 47 documents in which algorithms, an algorithm's related tagged lines and complexities coexist.

Reference Document Preparation. A reference document is prepared manually, by using those 47 documents, in which both algorithms and complexities are found. 471 relations are identified between algorithms, and their time and space complexities in 35 documents out of 47 documents. This dataset is used for results, comparisons and calculations.

Frequent Keywords Set. Frequent Keywords (FK) are extracted from algorithmic metadata lines and are used in matching the synopsis of the complexity line and the algorithmic metadata. The weightage of frequent keywords is less than normal keywords, as given in the following Inequality 1:

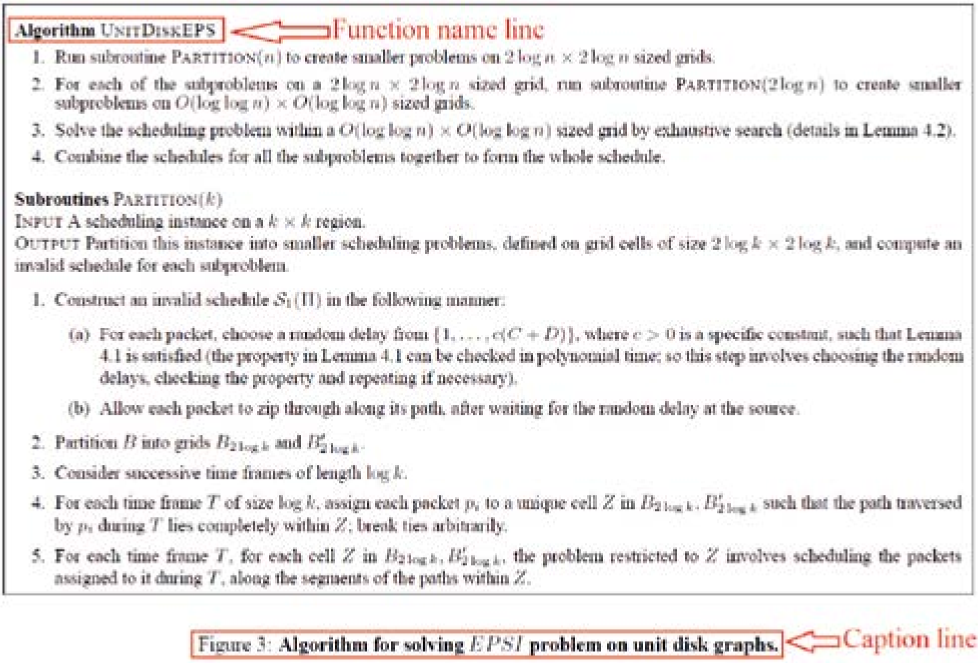
$$W_f < W_n \quad (1)$$

Where W_f is frequent keywords and W_n is normal or non-frequent keywords.

As frequent keywords are those keywords which are used more commonly than other, normal keywords, they have a smaller relative impact on finding

[10] The data and code used in this research can be downloaded from the following URL: https://github.com/slab-itu/icadl_link_algo.

the relevance of complexity context and algorithmic metadata. The list of frequent keywords is given in Table 1.



```
Algorithm UnitDiskEPS
  1. Run subroutine Partition($n$) to create smaller problems on $2 \log n \times 2 \log n$ sized grids.
  2. For each of the subproblems on a $2 \log n \times 2 \log n$ sized grid, run subroutine Partition($2 \log n$) to create smaller
     subproblems on $O(\log \log n) \times O(\log \log n)$ sized grids.
  3. Solve the scheduling problem within a $O(\log \log n) \times O(\log \log n)$ sized grid by exhaustive search (details in Lemma 4.2).
  4. Combine the schedules for all the subproblems together to form the whole schedule.

Subroutines Partition($k$)
Input A scheduling instance on a $k \times k$ region.
Output Partition this instance into smaller scheduling problems, defined on grid cells of size $2 \log k \times 2 \log k$, and compute an
invalid schedule for each subproblem.

  1. Construct an invalid schedule $S_1(\Pi)$ in the following manner:
      (a) For each packet, choose a random delay from $\{1, \dots, c(C + D)\}$, where $c > 0$ is a specific constant, such that Lemma
          4.1 is satisfied (the property in Lemma 4.1 can be checked in polynomial time; so this step involves choosing the random
          delays, checking the property and repeating if necessary).
      (b) Allow each packet to zip through along its path, after waiting for the random delay at the source.
  2. Partition $B$ into grids $B_{2 \log k}$ and $B'_{2 \log k}$.
  3. Consider successive time frames of length $\log k$.
  4. For each time frame $T$ of size $\log k$, assign each packet $p_i$ to a unique cell $Z$ in $B_{2 \log k}, B'_{2 \log k}$ such that the path traversed
     by $p_i$ during $T$ lies completely within $Z$; break ties arbitrarily.
  5. For each time frame $T$, for each cell $Z$ in $B_{2 \log k}, B'_{2 \log k}$, the problem restricted to $Z$ involves scheduling the packets
     assigned to it during $T$, along the segments of the paths within $Z$.
```

Figure 3: **Algorithm for solving $EPSI$ problem on unit disk graphs.**

Figure 5: Algorithmic Metadata Lines
(Kumar, Marathe, Parthasarathy and Srinivasan, 2004)

Cue words. We have used two lists of cue words (CW), one for complexity context and the other for the identification of common asymptotic growth rate function names. Cue words for complexity context are listed in Table 1. Cue words to identity common functions growth rate of asymptotic bounds are also listed in Table 1. Cue words are used to weigh the comparison between complexity and the algorithm; they are also used to identify the asymptotic function names of complexity.

WordNet Library. We used WordNet[11] library in Python for synonyms and semantically related terms along with original keywords to compare the context of the paragraph in which complexity is discussed and the algorithmic metadata, such as caption lines, indication sentences, function names or algorithm labels.

[11] https://wordnet.princeton.edu.

Table 1: Frequent Keywords Set, Cue Words for Complexity Context and for Common Complexity Functions

Sr.	Frequent Keyword	Cue Word for Complexity Context	Cue Word for Common Complexity Functions
1	algorithm	algorithm	polynomial
2	figure	complexity	poly
3	procedure	time complexity	constant
4	fig	space complexity	linear
5	method	run time	sublinear
6	following	best case	quadratic
7	steps	worst case	cubic
8	code	pseudocode	logarithmic
9	program	computational time	linearithmic
10	class	efficiency	exponential
11	function	optimal solution	parallel
12	search	performance	factorial
13	example		approximation
14	based		
15	sequence		
16	given		
17	skeleton		
18	table		
19	using		
20	main		
21	set		
22	list		
23	pseudocode		
24	described		
25	problem		

3.1 Pre-Processing

There are certain limitations in our dataset, some of which are discussed in this section.

3.1.1 PDF to TXT conversion

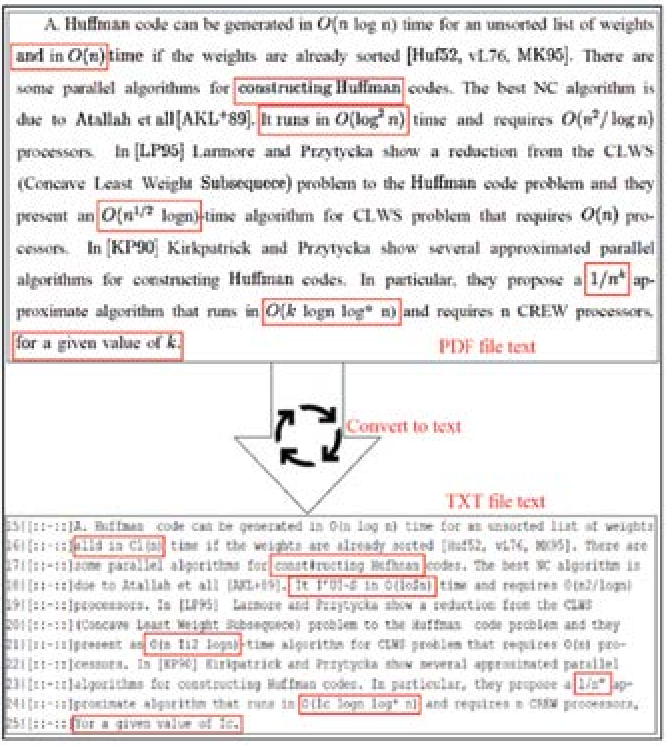
While extracting plain text from pdf documents, complex functions of complexity was not handled, and they are hard to identify in text document (as shown in Figure 6). Some words are not converted properly; most of them are special words such as the name of algorithm or keywords closely related to algorithmic metadata. The critical thing is that asymptotic notations are not converted properly, as can be observed in line 16 of the text document – "O" is translated to "Cl" – and because of this issue our regular expression will fail to identify the complexity line. If complexity line identification fails, then it will also fail to link that line to any algorithm as it is the base case for further processing.

3.1.2 Algorithmic Metadata Tagging

As we are using a dataset which is already tagged for algorithmic metadata, our results are dependent upon how accurately the metadata is tagged. Accuracy of algorithmic metadata tagging is 76% with 79% precision, 77% recall and 77% F1 scores from (Safder, Sarfraz, Hassan, Ali and Tuarob, 2017).

3.1.3 Multiple Complexity Lines Association

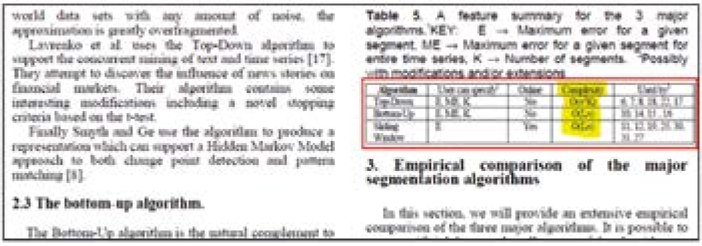
Sometimes multiple algorithms are described in a document or multiple versions of the same algorithm are given, and the complexity of all the algorithms or all versions of the algorithms are discussed in the same paragraph or sometimes in a table (as shown in Figure 7), so it is hard to distinguish which complexity line is related to which algorithm. For the tabular case, it may not associate any of the complexity lines to any algorithm because, when rendering the table from PDF to text, it may convert each cell of the table to a new line, and when we built the context of the complexity line, we only collect text from five lines before and after the complexity line. In the tabular case, this context will have only a few words, and this will not help to link it with algorithmic metadata. In this case, multiple complexity lines will be associated to an algorithm or multiple algorithms will be linked to same complexity line.



A. Huffman code can be generated in $O(n \log n)$ time for an unsorted list of weights and in $O(n)$ time if the weights are already sorted [Huf52, vL76, MK95]. There are some parallel algorithms for constructing Huffman codes. The best NC algorithm is due to Atallah et all[AKL+89]. It runs in $O(\log^2 n)$ time and requires $O(n^2/\log n)$ processors. In [LP95] Larmore and Przytycka show a reduction from the CLWS (Concave Least Weight Subsequece) problem to the Huffman code problem and they present an $O(n^{1/2} \log n)$-time algorithm for CLWS problem that requires $O(n)$ processors. In [KP90] Kirkpatrick and Przytycka show several approximated parallel algorithms for constructing Huffman codes. In particular, they propose a $1/n^k$ approximate algorithm that runs in $O(k \log n \log^* n)$ and requires n CREW processors, for a given value of k.

```
15|[::-::]A. Huffman  code can be generated in O(n log n) time for an unsorted list of weights
16|[::-::]alld in C1(n) time if the weights are already sorted [Huf52, vL76, MK95]. There are
17|[::-::]some parallel algorithms for const#ructing Huffman codes. The best NC algorithm is
18|[::-::]due to Atallah et all [AKL+89]. It I'UT-S in O(1ofn) time and requires O(n2/logn)
19|[::-::]processors. In [LP95]  Larmore and Przytycka show a reduction from the CLWS
20|[::-::](Concave Least Weight Subsequece) problem to the Huffman  code problem and they
21|[::-::]present an O(n 1:2 logn)-time algorithm for CLWS problem that requires O(n) pro-
22|[::-::]cessors. In [KP90] Kirkpatrick and Przytycka show several approximated parallel
23|[::-::]algorithms for constructing Huffman codes. In particular, they propose a 1/n* ap-
24|[::-::]proximate algorithm that runs in O(lc logn log* n) and requires n CREW processors.
25|[::-::For a given value of lc.
```

Figure 6: PDF to Text Conversion Issues (**Milidiú, Laber and Pessoa, 1999**)



world data sets with any amount of noise, the approximation is greatly overfragmented.

Lavrenko et al. uses the Top-Down algorithm to support the concurrent mining of text and time series [17]. They attempt to discover the influence of news stories on financial markets. Their algorithm contains some interesting modifications including a novel stopping criteria based on the t-test.

Finally Smyth and Ge use the algorithm to produce a representation which can support a Hidden Markov Model approach to both change point detection and pattern matching [8].

2.3 The bottom-up algorithm.

The Bottom-Up algorithm is the natural complement to

Table 5. A feature summary for the 3 major algorithms. KEY: E → Maximum error for a given segment, ME → Maximum error for a given segment for entire time series, K → Number of segments. *Possibly with modifications and/or extensions

Algorithm	User can specify	Online	Complexity	Used by*
Top-Down	E, ME, K	No	[illegible]	[illegible]
Bottom-Up	E, ME, K	No	[illegible]	[illegible]
Sliding Window	E	Yes	[illegible]	[illegible]

3. Empirical comparison of the major segmentation algorithms

In this section, we will provide an extensive empirical comparison of the three major algorithms. It is possible to

Figure 7: Complexities of Multiple Algorithms in a Table
(Keogh, Chu, Hart and Pazzani, 2001)

3.2 Error Rate

The error rate of our results may be high because the error will multiply with all error rates, such as the error rate of the pdf to text extraction, the error rate of the algorithmic metadata tagging and the error rate of our model. It can be calculated from the given Equation 2.

$$ER_{total}=ER_{pdfToText}\times ER_{tagging}\times ER_{ourModel} \quad (2)$$

where ER_{total} is the overall total error rate, $ER_{pdfToText}$ is the error rate of pdf to plain text extraction, $ER_{tagging}$ is the error rate of algorithmic metadata tagging and $ER_{ourModel}$ is the error rate of our model.

4. Methodology

Complexity lines are identified, and their context is built. Similarly, algorithmic metadata lines are extracted and combined for each algorithm. After that, by comparing both the complexity context and the algorithmic metadata, a reference file is created in which links between complexity lines and algorithms are listed. A high-level diagram of proposed system is given in Figure 8.

4.1 Complexity Line Identification and Context Building

We use regular expressions in Python to identify complexity lines in plain text documents, and asymptotic notation formats are used in the regular expressions for this purpose. After identification of complexity lines, we have built context of the line, which is identified as a complexity line, to build context – five lines before and after the complexity line are used. Tagged; lines for algorithmic metadata are ignored while building the context. There are multiple complexity lines in the same document, and context of each complexity line is built separately.

The grammar for our Regular Expression (RE) which is used to detect complexity lines from text documents is given in Figure 9, and the Python notation is given below:

r'\b\d*[OΩΘω0]\(.*[nmk\d(log)(ln)].*\)′ (3)

There are two main parts of this regular expression: the first part is to detect asymptotic bound notations such as O, Ω, Θ and ω; the second part is enclosed in starting parenthesis "(" and closing parenthesis ")", and between these there can be any complexity notation: *n* is mostly used for input size or data size and some other letters such as *m* and *k* are also used for the same purpose; \d is used for number in regular expressions, it is used here for power or constant value detection; sometimes complexity is defined in a logarithmic function, *(log)* and *(ln)* are used for logarithmic function detection. All these special characters and symbols are enclosed in a starting bracket "[" and a closing bracket "]" to ignore their sequence and occurrence; case is also ignored to detect upper- and lower-case letters.

4.2 Section Detection

The role of sections is very important when extracting relevant information. In scientific documents, sections can be identified using section headers and their boundaries (Tuarob, Mitra and Giles, 2015). In our case, if a complexity line lies in related work or background section, it may be related to an algorithm which is not discussed in the current document and the author may be comparing the complexities of different algorithms. If it is in the implementation or methodology section, then there are high chances that it is related to algorithms which are described in the current document. Similarly, if it is in the abstract, then chances are very high for that. If it is in future work, then it may be the desired complexity to achieve in future. If it is in a reference section then it will be ignored, because in this case it will be part of some other document's title.

4.3 Algorithm Metadata Extraction and Compilation

As algorithmic metadata is the lines related to an algorithm's description and definition, these lines have already been tagged in our dataset (*e.g.*, caption lines and algorithmic labels). These algorithmic metadata lines are extracted from the plain text document and combined together. Most frequent keywords are also calculated using the frequency of the keywords in all documents. In some documents there are multiple algorithms; the metadata of each of them is combined separately.

4.4 Comparison of an Algorithm's Metadata and Context of Complexity

We have built algorithmic metadata using tagged lines from plain text document, identified complexity lines and built their context from the same document, as shown in Figure 10. We then compare both of them and use a probabilistic method to compare algorithmic metadata and complexity line context. We have also used weights for direct keywords matching and synonyms and semantically related terms. Synonyms and semantically related terms have been extracted from WordNet library using Python.

Weights for direct keywords are higher than for synonyms and semantically related terms and weights for the most frequent terms are lower than for the less frequent terms in algorithmic metadata. By combining both measures, the following Inequality 4 is applied for matching keywords:

$$W1 > W2 > W3 > W4 \quad (4)$$

where W1 is for direct non-frequent keywords, W2 is for direct frequent keywords, W3 is for synonyms or semantically related non-frequent keywords and W4 is for synonyms or semantically related frequent keywords.

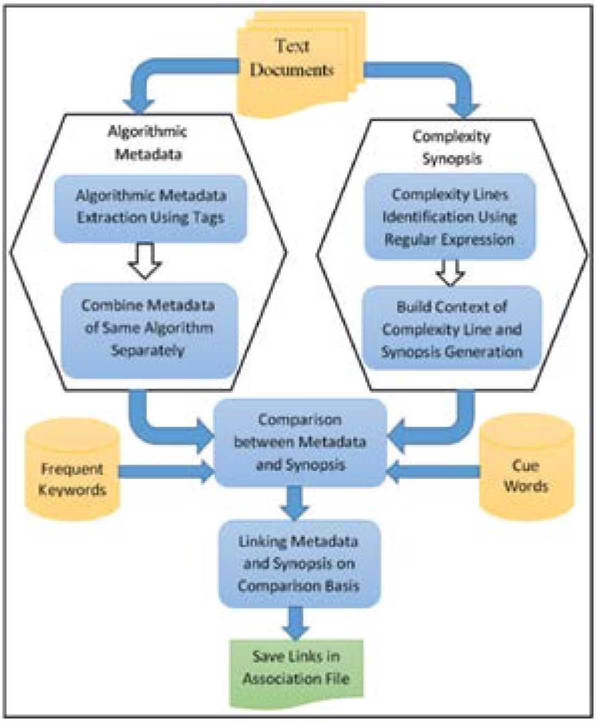
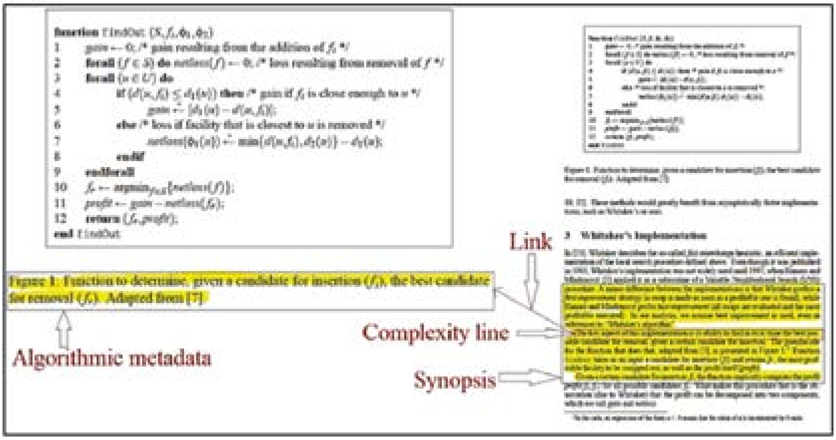

Text Documents
Algorithmic Metadata
Algorithmic Metadata Extraction Using Tags
Combine Metadata of Same Algorithm Separately
Complexity Synopsis
Complexity Lines Identification Using Regular Expression
Build Context of Complexity Line and Synopsis Generation
Frequent Keywords
Comparison between Metadata and Synopsis
Cue Words
Linking Metadata and Synopsis on Comparison Basis
Save Links in Association File

Figure 8: High-Level Diagram of Proposed System

```
    <ALGO_COMP>  ::= <ASYMP_NOTATIONS> <(> <COMP_TEXT> <)>
<ASYMP_NOTATIONS>::= O|Ω|Θ|o|ω
    <COMP_TEXT>  ::= <A String of Characters>
```

Figure 9: Grammar for Algorithmic Complexity



```
function findOut (S, f_i, φ1, φ2)
    gain ← 0; /* gain resulting from the addition of f_i */
    forall (f ∈ S) do netloss(f) ← 0; /* loss resulting from removal of f */
    forall (u ∈ U) do
        if (d(u, f_i) ≤ d1(u)) then /* gain if f_i is close enough to u */
            gain +← [d1(u) − d(u, f_i)];
        else /* loss if facility that is closest to u is removed */
            netloss(φ1(u)) +← min{d(u, f_i), d2(u)} − d1(u);
        endif
    endforall
    f_r ← argmin_{f∈S}{netloss(f)};
    profit ← gain − netloss(f_r);
    return (f_r, profit);
end findOut
```

Figure 1: Function to determine, given a candidate for insertion (f_i), the best candidate for removal (f_r). Adapted from [7]

3 Whitaker's Implementation

Figure 10: Comparison of Algorithm's Metadata and Context of Complexity

4.5 Reference or Association File for Algorithm and Complexity Relations

A reference file is created to save the links between complexities and algorithms. As an algorithm can be linked to more than one complexity line and one complexity line can be associated with more than one algorithm, we have created a dataset for associations or links between algorithmic metadata and complexity lines. In this dataset we have saved the line numbers of the complexity lines with the text of the line and the line numbers of the algorithmic metadata with the metadata itself. Some other fields have been added in this dataset, such as number of keywords in algorithmic metadata, percentage of matching keywords and cue words that are matched in complexity context (both for complexity context and asymptotic function). We have saved this dataset to a file called reference or association file and this file will be used for ranking and indexing the algorithms for IR systems.

5. Experiments and Results

We have done a number of experiments on data to improve our results by changing matching percentage of algorithmic metadata with complexity synopsis-built form complexity context and with and without considering frequent keywords set and cue words for complexity context and asymptotic growth function names. We have also selected some feasible thresholds using experimental results to improve our results. Some of these experiments and their results are discussed in this section. Graphs for threshold values selection, ROC curves and precision-recall covers are also given.

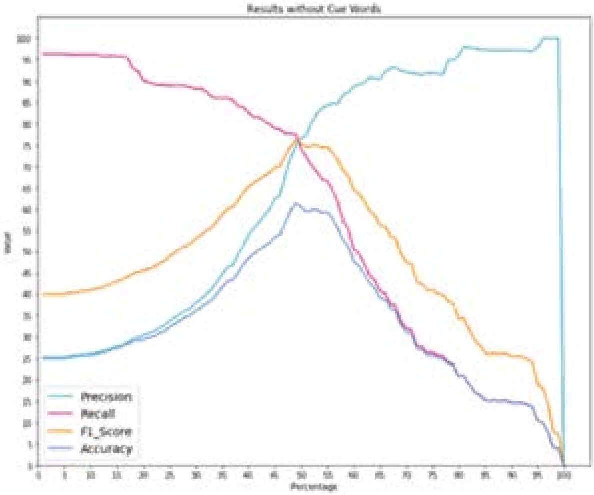


Figure 11: Results without Cue Words for Threshold Selection

5.1 Threshold Selection

Thresholds for matching the percentage ratio of algorithmic metadata and the complexity synopsis are selected using precision, recall, F1-score and accuracy data for all threshold values from 0 to 100 percent.

Graphs for all thresholds are also shown in Figure 11 and Figure 12. In both figures recall is very high at the beginning but goes down as we increase the value of the matching percentage; for the precision value it is low at the beginning but grows when we increase the value of the matching percentage. It is because almost all true links are identified when the percentage is low, but the false links identification ratio is also very high at a low matching percentage. Similarly, at a very high matching percentage only a few true links are identified but the false links identification ratio is also negligible. At the point where precision and recall curves intersect, the ratio of true prediction is maximum.

F1-score and accuracy are low at the beginning, then they grow up to some point (near the intersection of precision and recall curves), and then they go down again, because F1-score is the harmonic average of precision and recall and accuracy is the ratio of correctly identified instances to the total number of instances.

Percentage threshold without using cue words is selected as 50 percent by using results of precision, recall, F1-score and accuracy, as shown in Figure 11, in which the intersection of precision, recall and F1- score curves is near 50 percent. Similarly, percentage threshold for with using cue words is selected as 55 percent by using results of precision, recall, F1-meaure and accuracy, as shown in Figure 12. Similarly in this figure the intersection of precision, recall and F1-score curves is near 55 percent. These thresholds are used in our experimental setup, which is given in the next sections.

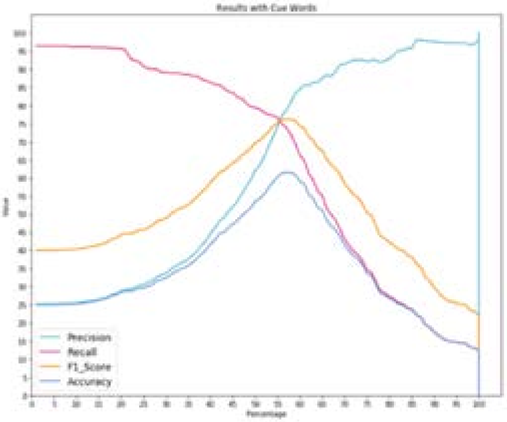


Figure 12: Results with Cue Words Threshold Selection

5. 2 Experimental Setup

We always learn from our experiments, and we have done several experiments on our data to improve our results, but as we discussed earlier, there are some limitations in our data and model, so we can only achieve our results up to some possible level. Four experiments are discussed in the following sections.

Table 2: Truth Table, 50% Matched, with No-Frequent Keywords and No-Cue Words (NFNC50)

Measures	**Value**
Total True	471
True matched	365
False matched	189
True and False Both matched	554
Not matched	106
Total	660

5.2.1 Experiment 1

In the first experiment, we used a matching percentage ratio of algorithmic metadata up to 50 percent, and in this experiment, we completely ignored the cue words and the frequent keywords from the algorithmic metadata. We named this experiment NFNC50 (No-Frequent keywords and No-Cue words with 50% threshold). Results for experiment 1 are summarized in Table 2.

5.2.2 Experiment 2

In the second experiment, we used a matching percentage ratio of algorithmic metadata up to 50 percent, and in this experiment we ignored the cue words, but we used frequent keywords from the algorithmic metadata. Frequent keywords are considered as low weighted in this case, frequent keywords set is generated from algorithmic metadata as discussed in Section 3.1.2. We named this experiment FNC50 (Frequent keywords and No-Cue words with 50% threshold). Results for experiment 2, are summarized in Table 3.

Table 3: Truth Table, 50% Matched, with Frequent Keywords and No-Cue Words (FNC50)

Measures	Value
Total True	471
True matched	359
False matched	104
True and False Both matched	463
Not matched	112
Total	575

5.2.3 Experiment 3

In this experiment, we used matching percentage ratio of algorithmic metadata along with cue words, greater than 55 percent, and, we added the cue words matching ratio for the overall percentage. Frequent keywords are also considered low weighted in this case. We named this experiment FC55 (Frequent keywords and Cue words with 55% threshold). Results for this experiment are summarized in Table 4.

Table 4: Truth Table, 50% Metadata Matched Overall with Cue Words (FC55)

Measures	Value
Total True	471
True matched	355
False matched	111
True and False Both matched	466
Not matched	116
Total	582

5.2.4 Experiment 4

In this experiment, we used a matching percentage ratio of algorithmic metadata up to 50 percent separately, and we added the cue words matching ratio after that. In other words, we combined the conditions of the second and the third experiment, and in this way, the results were maximized. We named this experiment FNC50FC55. Results for this experiment are summarized in Table 5.

Table 5: Truth Table, 50% Metadata Matched and 55% Overall with Cue Words (FNC50FC55)

Measures	**Value**
Total True	471
True matched	354
False matched	86
True and False Both matched	440
Not matched	117
Total	557

5.3 Calculations

Results are calculated using standard formulas as discussed in this section.

5.3.1 Precision

Precision is defined as the ratio of correctly identified instances to the total predicted positive instances. The equation to calculate precision is given as follows:

$$Precision = \frac{\text{True matched (TP)}}{\text{True and False Both Matched (TP+FP)}} \qquad (5)$$

The worst precision is recorded for experiment 3 (FC55), which is 61 percent, and the best precision is yielded by experiment 4 (FNC50FC55), which is 81 percent.

5.3.2 Recall

Recall is defined as the ratio of correctly identified instances to all actual observations in the data; it is also called 'sensitivity'. Recall is calculated using the following equation:

$$Recall = \frac{\text{True matched (TP)}}{\text{Total True (TP+FN)}} \qquad (6)$$

Experiment 1 (NFNC50) yielded the best recall, which is 77 percent, and the worst recall is recorded for experiment 3 (FC55), which is 75 percent.

F1 Score

It is the harmonic average of precision and recall, as given below:

$$\text{F1 Score} = \frac{2*(Recall * Precision)}{(Recall + Precision)} \qquad (7)$$

F1 score is more useful than accuracy. The best F1 score is yielded by experiment 4 (FNC50FC55), which is 78 and the worst F1 score is yielded by experiment 3 (FC55), which is 67 percent.

Accuracy

Accuracy is the most common measure to check the performance of results; it is the ratio of correctly identified instances to the total number of instances, as given below:

$$\text{Accuracy} = \frac{\text{True matched (TP+TN)}}{\textit{Total (TP+FP+FN+TN)}} \qquad (8)$$

The best accuracy is yielded by experiment 4 (FNC50FC55), which is 65 and the worst accuracy is yielded by experiment 1 (NFNC50), which is 56 percent.

Results

Results are shown in Table 6, as in our first experiment we did not use frequent keywords from algorithmic metadata and cue words for complexity context and asymptotic function names are completely ignored. Recall was maximum, which is 77%, but accuracy is low. In other words, in this case maximum actual links are identified but false results ratio is also high.

In the second experiment we have considered only frequent keywords for matching weights and we see that results are improved, as precision was improved from 66% to 78% and accuracy – from 56% to 64%; however, recall was down from 77% to 76%,

In our third experiment, we have also considered the cue words along with frequent keywords to evaluate the weights and increase the matching percentage threshold from 50% to 55%. In this case results were not improved, as we can see in the third row of Table 6.

In our last experiment, we have combined the conditions and thresholds of the second and third experiment. By doing this we got maximum results, as precision is improved significantly; F1 score and accuracy is also maximized in this case. Finally, we have achieved 81% precision, 75% recall, 78% F1-score and 65% accuracy.

Precision, recall, F-measure and accuracy for linking the algorithm and complexity line for the different experiments are given in Table 6.

Table 6: Precision, Recall, F-measure and Accuracy for Algorithm and Complexity Line Linking

Name	Method	Precision	Recall	F1 Score	Accuracy
NFNC50	50% matched with no frequent keywords and no cue words	0.66	**0.77**	0.71	*0.56*
FNC50	50% matched with frequent keywords and no cue words	0.78	0.76	0.77	0.64
FC55	55% matched metadata and cue words overall	*0.61*	*0.75*	*0.67*	0.62
FNC50FC55	Combination of the second and the third experiment	**0.81**	0.75	**0.78**	**0.65**

5.4.1 ROC and Precision-Recall Curves

Receiver Operating Characteristic (ROC) curves are shown in Figure 13, in which we can see that the area under the curve for NFNQ50 is 0.90, which is the maximum among the other curves because recall for this experiment was maximum.

Precision-Recall curves are shown in Figure 14. As can be seen that the areas under curves for all three experiment 2, 3 and 4 are almost same because recall for these experiments is almost the same, and the area under the curve for experiment 1 (NFNQ50) is the minimum.

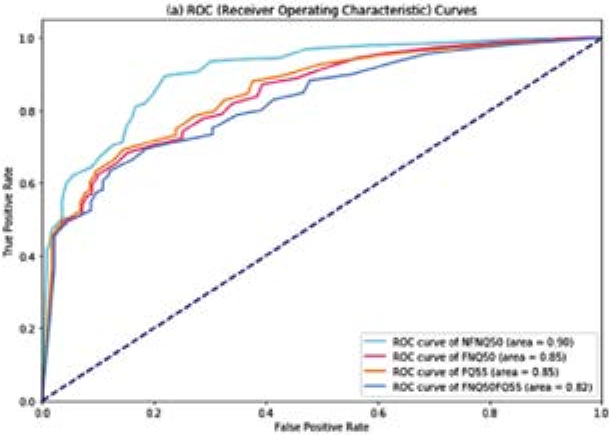


Figure 13: ROC (Receiver Operating Characteristic) Curves

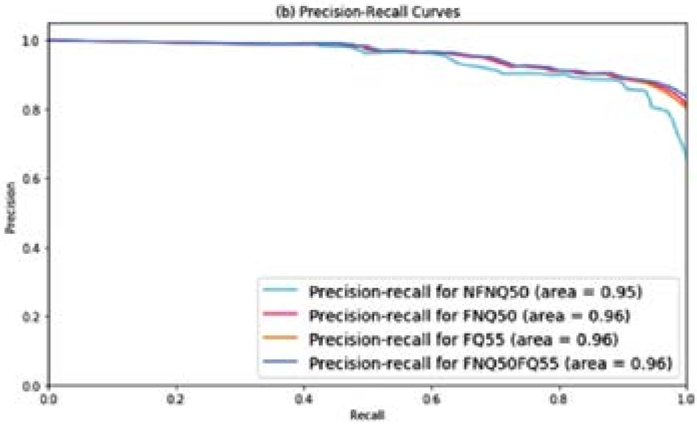


Figure 14: Precision-Recall Curves

6. Conclusion and Future Work

6.1 Concluding Remarks

Linking non-textual document elements (NTDEs), such as charts, diagrams, pseudocodes and tables to their relevant paragraph in a scholarly document is a critical process to improve the relevant results for IR systems which are mainly focused on this type of search. In this research we have focused on algorithms and their complexities linking using complexity lines synopsis.

Scientific publications host a tremendous number of high-quality algorithms developed by professional researchers. In this paper we have linked the complexity lines and the algorithmic metadata in the same scientific document. We have used keywords from algorithmic metadata and a synopsis generated from five lines before and after the complexity line. Complexity of algorithms, for both time and space, is the main concern of developers and researchers. Currently, IR systems for algorithmic search did not directly consider relevant complexity of algorithms to rank and order the results.

In our research, a frequent keywords set and two cue words sets have been used to improve our results. Precision, recall, F1-meaure and accuracy graphs have been used for thresholds selections. Complexity lines have been identified by regular expressions with the use of asymptotic notations. WordNet library has been used for synonyms or related terms. A reference file has been manually annotated for results and comparisons. An associated file has been created to save the links between the complexity lines and their corresponding algorithms. A number of experiments have been performed with different combinations of frequent keywords, cue words, synonyms or related terms, and thresholds selections for metadata comparison. We have improved our results by combining the best performing experiments.

Future Work

In the future we can use similar linking methodology to link different non-textual document elements, such as figures, tables and charts to their relevant paragraphs in the same document. By using our methodology, we can extract and catalogue relevant algorithms and can introduce several exciting applications including discovering new or enhanced algorithms or analysing different versions of an algorithm. We can also improve algorithmic information retrieval systems by using the complexity of algorithms to index and rank the algorithmic search results. An AI enabled search engine architecture is used in (Safder, Hassan and Aljohani, 2018) and (Safder and Hassan, 2018), where an EMD embedded model is used to improve relevant information retrieval, which is a RCNN based model; we can use our algorithmic metadata and complexity lines context to improve this search engine.

References

Al-Zaidy, R. A. and Giles, C. L. (2017). A Machine Learning Approach for Semantic Structuring of Scientific Charts in Scholarly Documents. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(2), 4644–4649. Avajlable at: https://doi.org/10.1609/aaai.v31i2.19088.

Bajracharya, S., Ossher, J. and Lopes, C. (2009). Sourcerer: An Internet-scale Software Repository. In: *Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*. IEEE Computer Society, 1–4.

Baker, J. B., Sexton, A. P., Sorge, V. and Suzuki, M. (2011). Comparing Approaches to Mathematical Document Analysis from PDF. In: *International Conference on Document Analysis and Recognition*. IEEE, 463–467.

Bhatia, S. and Mitra, P. (2012). Summarizing Figures, Tables, and Algorithms in Scientific Publications to Augment Search Results. *ACM Transactions on Information Systems (TOIS),* 30(1), 3.

Bhatia, S., Mitra, P. and Giles, C. L. (2010). Finding Algorithms in Scientific Articles. In: *Proceedings of the 19th International Conference on World Wide Web*. NY: Association for Computing Machinery (ACM), 1061–1062.

Bhatia, S., Tuarob, S., Mitra, P. and Giles, C. L. (2011). An Algorithm Search Engine for Software Developers. In: *Proceedings of the 3rd International Workshop on Search-Driven Development: Users, Infrastructure, Tools, and Evaluation*. NY: ACM, 13–16.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research,* 3(Jan), 993–1022.

Chen, H.-H., Gou, L., Zhang, X. and Giles, C. L. (2011). Collabseer: A Search Engine for Collaboration Discovery. In: *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*. NY: ACM, 231–240.

Chen, P., Xie, H., Maslov, S. and Redner, S. (2007). Finding Scientific Gems with Google's PageRank Algorithm. *Journal of Informetrics,* 1(1), 8–15.

Coüasnon, B. and Lemaitre, A. (2014). Recognition of Tables and Forms. In: *Handbook of Document Image Processing and Recognition*. London: Springer, 647–677.

Cormen, T. H. (2013). *Algorithms Unlocked.* The MIT Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009). *Introduction to Algorithms.* 3rd ed. The MIT Press.

Elminaam, D. S., Abdual-Kader, H. M. and Hadhoud, M. M. (2010). Evaluating the Performance of Symmetric Encryption Algorithms. *IJ Network Security,* 10(3), 216–222.

Hearst, M. A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M. A. and Ye, J. (2007). BioText Search Engine: Beyond Abstract Search. *Bioinformatics,* 23(16), 2196–2197.

Hirschberg, D. S. (1975). A Linear Space Algorithm for Computing Maximal Common Subsequences. *Communications of the ACM,* 18(6), 341–343.

Jung, E., Elmallah, E. S. and Gouda, M. G. (2007). Optimal Dispersal of Certificate Chains. *IEEE Transactions on Parallel and Distributed Systems,* 18(4), 474–484.

Keogh, E., Chu, S., Hart, D. and Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series. In: *Proceedings 2001 IEEE International Conference on Data Mining.* IEEE, 289–296.

Khabsa, M. and Giles, C. L. (2014). The Number of Scholarly Documents on the Public Web. *PloS one,* 9(5), e93949. Available at: https://doi.org/10.1371/journal.pone.0093949.

Khabsa, M., Treeratpituk, P. and Giles, C. L. (2012). Ackseer: A Repository and Search Engine for Automatically Extracted Acknowledgments from Digital Libraries. In: *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*. NY: ACM, 185–194. .

Khan, S., Liu, X., Shakil, K. A. and Alam, M. (2017). A Survey on Scholarly Data: From Big Data Perspective. *Information Processing & Management,* 53(4), 923–944.

Kim, H.-S., Lee, J.-H. and Jeong, Y.-S. (2003). Method for Finding Shortest Path to Destination in Traffic Network Using Dijkstra Algorithm or Floyd-warshall Algorithm. Google Patents.

Kleinberg, J. and Tardos, É. (2009). *Algorithm Design.* Boston: Pearson/Addison-Wesley.

Kumar, V., Marathe, M. V., Parthasarathy, S. and Srinivasan, A. (2004). End-to-end Packet-scheduling in Wireless Ad-hoc Networks. In: *Proceedings of the Fifteenth*

Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 1021–1030.

Lai, S., Xu, L., Liu, K. and Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), 2267–2273. Avajlable at: https://doi.org/10.1609/aaai.v29i1.9513.

Li, H. (2013). Aligning Sequence Reads, Clone Sequences and Assembly Contigs with BWA-MEM. *arXiv.org e-Print arXiv:1303.3997*. Available at: https://doi.org/10.48550/arXiv.1303.3997.

Liu, Y., Bai, K., Mitra, P. and Giles, C. L. (2007). Tableseer: Automatic Table Metadata Extraction and Searching in Digital Libraries. In: *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*. NY: ACM, 91–100.

Marron, M., Stefanovic, D., Hermenegildo, M. and Kapur, D. (2007). Heap Analysis in the Presence of Collection Libraries. In: *Proceedings of the 7th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*. NY: ACM, 31–36.

McMillan, C., Grechanik, M., Poshyvanyk, D., Fu, C. and Xie, Q. (2012). Exemplar: A Source Code Search Engine for Finding Highly Relevant Applications. *IEEE Transactions on Software Engineering,* 38(5), 1069–1087.

Milidiú, R. L., Laber, E. S. and Pessoa, A. A. (1999). A Work-efficient Parallel Algorithm for Constructing Huffman Codes. In: *Data Compression Conference, 1999. Proceedings. DCC'99*. IEEE, 277–286.

Nadeem, A. and Javed, M. Y. (2005). A Performance Comparison of Data Encryption Algorithms. In: *2005 First International Conference on Information and Communication Technologies (ICICT)*. IEEE, 84–89.

Nargesian, F., Zhu, E., Pu, K. Q. and Miller, R. J. (2018). Table Union Search on Open Data. In: *Proceedings of the VLDB Endowment,* 11(7), 813–825.

Ratliff, N. D. and Bagnell, J. A. (2007). Kernel Conjugate Gradient for Fast Kernel Machines. In: *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, 1017–1022.

Safder, I. and Hassan, S.-U. (2018). DS4A: Deep Search System for Algorithms from Full-text Scholarly Big Data. In: *International Conference on Data Mining Workshop (ICDMW),* 1308–1315.

Safder, I., Hassan, S.-U. and Aljohani, N. R. (2018). AI Cognition in Searching for Relevant Knowledge from Scholarly Big Data, Using a Multi-layer Perceptron and Recurrent Convolutional Neural Network Model. In: *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 251–258.

Safder, I., Sarfraz, J., Hassan, S.-U., Ali, M. and Tuarob, S. (2017). Detecting Target Text Related to Algorithmic Efficiency in Scholarly Big Data using Recurrent Convolutional Neural Network Model. In: Choemprayong, S., Crestani, F., Cunningham, S., (eds.). *Digital Libraries: Data, Information, and Knowledge for Digital Lives. ICADL 2017.* Cham: Springer, 30–40.

Siegel, N., Horvitz, Z., Levin, R., Divvala, S. and Farhadi, A. (2016). FigureSeer: Parsing Result-figures in Research Papers. In: Leibe, B., Matas, J., Sebe, N., Welling, M., (eds.). *Computer Vision – ECCV 2016,* 9911, 664–680. Available at: https://link.springer.com/chapter/10.1007/978-3-319-46478-7_41.

Siegel, N., Lourie, N., Power, R. and Ammar, W. (2018). Extracting Scientific Figures with Distantly Supervised Neural Networks. In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. NY: ACM, 223–232.

Stewart, J. G. (2009). Genre Oriented Summarization. [PhD diss.]. Carnegie Mellon University, Language Technologies Institute, School of Computer Science.

Ochieng, P. J., Djatna, T. and Kusuma, W. A. (2015). Tandem Repeats Analysis in DNA Sequences Based on Improved Burrows-Wheeler Transform. In: *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 117–122.

Tuarob, S., Bhatia, S., Mitra, P. and Giles, C. L. (2013). Automatic Detection of Pseudocodes in Scholarly Documents Using Machine Learning. In: *12th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 738–742.

Tuarob, S., Bhatia, S., Mitra, P. and Giles, C. L. (2016). AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data. *IEEE Transactions on Big Data,* 2(1), 3–17.

Tuarob, S., Mitra, P. and Giles, C. L. (2015). A Hybrid Approach to Discover Semantic Hierarchical Sections in Scholarly Documents. In: *13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1081–1085.

Tyagi, N. and Sharma, S. (2012). Weighted Page Rank Algorithm Based on Number of Visits of Links of Web Page. *International Journal of Soft Computing and Engineering (IJSCE),* 2(3), 2231–2307. Available at: https://www.ijsce.org/wp-content/uploads/papers/v2i3/C0796062312.pdf.

Wang, J. (2009). Mean-variance Analysis: A New Document Ranking Theory in Information Retrieval. *European Conference on Information Retrieval.* Springer, 4–16.

Wise, M. J. (1995). Neweyes: A System for Comparing Biological Sequences Using the Running Karp-Rabin Greedy String-Tiling Algorithm. In: *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 3, 393–401.

Wu, J., Williams, K. M., Chen, H.-H., Khabsa, M., Caragea, C., Tuarob, S., Ororbia, A., D Jordan, D. and Giles, C. L. (2015). Citeseerx: AI in a Digital Library Search Engine. *AI Magazine,* 36(3), 35–48.

Yang, Y., Yu, P. and Gan, Y. (2011). Experimental Study on the Five Sort Algorithms. In: *2011 Second International Conference on Mechanic Automation and Control Engineering (MACE)*. IEEE, 1314–1317.

Zanibbi, R. and Blostein, D. (2012). Recognition and Retrieval of Mathematical Expressions. *International Journal on Document Analysis and Recognition (IJDAR),* 15(4), 331–357.

https://doi.org/10.33919/JCAL.23.1.3

COMPUTER-ASSISTED TRANSCRIPTION AND ANALYSIS OF BULGARIAN CHILD SPEECH DATA USING CHILDES AND CLAN

Velka Popova[1], Dimitar Popov[1]

[1] *Konstantin Preslavsky University of Shumen*

Abstract

The present paper focuses on the possibilities offered by corpus linguistics in the study of child speech, with its specificities as a linguistic phenomenon. An attempt is made to highlight the advantages of the CHILDES system for studying spontaneous speech interaction in the Bulgarian corpus of child language (Bulgarian LabLing Corpus), in which the data are transcribed and annotated within this paradigm.

Keywords: *CHILDES, CLAN, Bulgarian LabLing Corpus*

1. Introduction

In recent decades, linguistic resources organised as corpora, have been increasingly used in the modelling of language and speech behaviour of its speakers, despite the fact that the creation and maintenance of computerised corpora is extremely laborious and costly. Modern technologies required a new, more effective standard for data presentation and processing. They made it possible to extend the scope of a corpus to millions of language items while also optimising the options for their annotation (linguistic analysis), unification, standardisation and repeated use. The magnitude of change in research is even more evident in the application of modern computer programmes for automatic processing of huge databases in the corpus approach to research in the field of humanities.

The present paper focuses on the possibilities offered by corpus linguistics in the study of child speech, with its specificities as a linguistic phenomenon. An attempt is made to highlight the advantages of the CHILDES system for studying spontaneous speech interaction in the Bulgarian corpus of child language (Bulgarian LabLing Corpus), in which the data are transcribed and annotated within this paradigm. The corpus is available to researchers at:

https://childes.talkbank.org/access/Slavic/Bulgarian/LabLing.html.

2. Corpus paradigm – necessity or fad in child language research

The proposed study addresses the question of whether the use of a corpus paradigm in studying child language is a necessity or a fad, having in mind that the creation of computerised corpora is an extremely costly and laborious endeavour. There is the question if child language as a specific linguistic phenomenon is worthy of studying by means of sophisticated research tools. This in turn leads to the question whether we need to study child language at all, and what the advantages of the corpus paradigm are in comparison to some traditional approaches which have been used so far in the study of linguistic ontogenesis.

In the linguistics tradition there has been an enduring interest in the phenomenon of child language and this is not only for the sake of studying it or out of mere research curiosity. On the contrary, data about linguistic ontogenesis are in many cases the mandatory 'external evidence' for testing different hypotheses or theoretical constructs. Along with their importance for clarifying issues in linguistic typology and universals, these data are crucial in resolving problems of early speech pathology and language teaching. In recent decades, in line with the fast developments in psycholinguistics and cognitive linguistics, child language has also proved to be the key to the hidden functioning of the human perceptive and cognitive faculties. In this way, the research of linguistic ontogenesis is part of the general tendency in modern linguistics to overcome isolated study of language for the sake of it and focus on human speech interaction.

The importance of child language necessitates the creation of an adequate model of language ontogenesis, which in turn raises the question of the suitability and (in)sufficiency of the approaches which can help debunk myths not only in linguistics, but also in psycholinguistics, and in the theory of language learning. This in turn requires the use of relevant models and reliable empirical material.

Next comes the question of the methods for collecting sufficient quantitative and qualitative empirical evidence for adequate study of child language. In this regard, it should be noted that ever since Charles Darwin, the corpus approach has been a major factor in the research on language ontogenesis. There is ample evidence in support of such an approach.

Developments in technology over time have brought about a new quality of empirical data and the possibilities for their processing. Files and diaries have been replaced with electronic data of speech recordings, and the hard, intensive and exhausting work involved in the registration, transcription and statistical processing of data is now replaced by modern technology and software products. The apogee of this evolutionary process is the creation of the CHILDES system. The typological diversity of the included linguistic data, the unified manner of transcription, and the package of programme resources CLAN for automatic processing, turn this system into an extremely useful and convenient research platform. In the following section of this paper we will use the example of the Bulgarian CHILDES-Corpus to demonstrate its tools for empirical verification of the models of language ontogenesis. We will look for an answer to the question of the role of the computerised CHILDES system in overcoming difficulties associated with the specifics of child language.

3. Bulgarian resources of spontaneous child speech in CHILDES terminology

In the autumn of 2020, a new addition appeared in the database of the CHILDES platform in its Slavic languages section, namely the *Bulgarian LabLing Corpus*. It appeared as a result of long years of work done by researchers from LABLING. The corpus data are transcribed in the unified CHAT format of the CHILDES system (MacWhinney, 2010), which makes them comparable to the corpora in other languages in the platform. Long before the publication of the Bulgarian LabLing Corpus, the application and reliability of this base comprising speech data and information obtained from Bulgarian children was partially approbated in discussions and comparative analyses of Bulgarian and the other languages (in particular, German and

Russian), carried out in the sphere of cross-linguistic programme for examining the early adoption and mastering of the aspect (comp.: Kühnast et al., 2004; Bittner et al., 2005). The corpus also stresses the empirical base and the multiplicity of particular research works on different aspects of the early-age ontogenesis of Bulgarian grammar. Bulgarian computarised empirical data have been used in the process of the empirical verification of the pre- and proto-morphology model (see: Popova, 2007; Popova, 2016).

The present study is based on the longitudinal subcorpus of the Bulgarian LabLing corpus. This collection comprises spontaneous speech samples produced by five children aged between 1 and 3 years. In the core of the database there are 47 hours of recordings. They are presented on the CHILDES platform in 104 files in CHAT-format.

4. How does CHILDES provide sufficient and reliable information needed for linguistic analysis?

The Bulgarian child language corpus was created by using the two main tools of the CHILDES system: the special format for transcribing and coding - CHAT, and the package of programmes for analysis - CLAN.

The file format in CHILDES is called CHAT and all files are saved as *.cha. For transcription and playback, the relevant part of CLAN is the editor. The editor uses most of the same conventions as Microsoft Word. However, unlike Word, it allows the researcher to link individual segments of the transcript directly to the audio or video media.

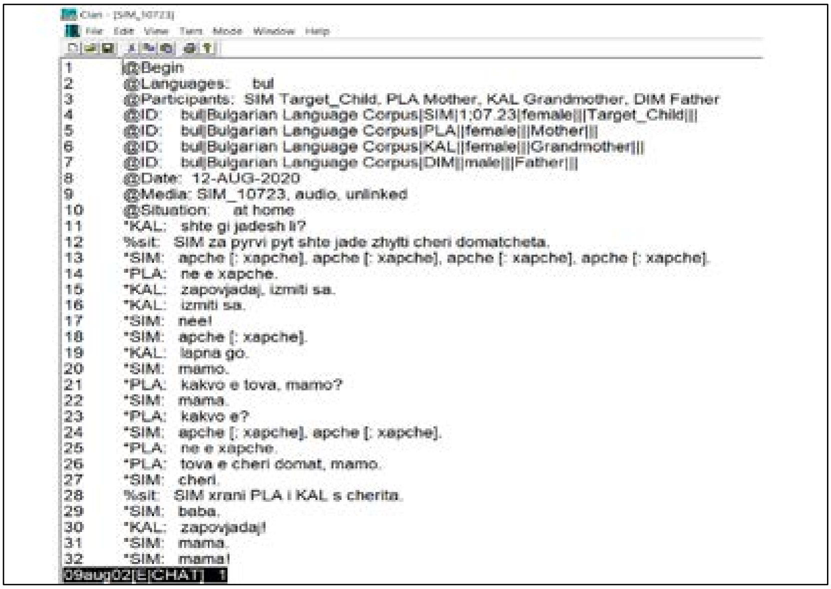
Within CHILDES the Bulgarian resources of spontaneous child speech are presented in the mandatory CHAT format (MacWhinney, 2000). It includes the following: 1. Title lines, containing information about the participants in the dialogue, their age, date of birth, date and conditions of the recording; 2. The alternating utterances of the participants, formed as single lines and the accompanying comments, given as additional lines. The headings were chosen by the researcher, and @Begin, @Participants and @End were left as mandatory, as in the following sample transcript:

```
@Begin
@Participants:      ALE Alexandra Target_Child, VEL Velka Mother
@ALE's birthdate:  29-JAN-1989
@Date:       27-MAR-1990
@Filename:          al10129
@Age of ALE:        1;01.29
@Situation:         at home
```

```
*VEL:	[spoken material]
*ALE:	[spoken material]
*VEL:	[spoken material]
@End
```

For modern studies of early linguistic ontogenesis in the context of constructivism, it is particularly important that the data not only of children but also of adults who care for them is taken into consideration. With the CHILDES standard for transcription, optimal conditions are created for them to be adequately described, as it includes mandatory order Participants. Here, together with their names and social roles, a three-letter code is introduced, which starts the lines of each participant in the subsequent dialogue. Thus, optimal conditions are created to isolate, monitor and analyse the lines of each participant, which would lead to a more objective study of child speech and child-directed speech.

In this regard, a wide applicability of the Longitudinal Corpus of the Bulgarian CHILDES collection can be expected, as each of the transcripts includes data on the identification of the participants (demographic and linguistic parameters) and the respective corpus. See Fig. 1:



```
@Begin
@Languages:	bul
@Participants:	SIM Target_Child, PLA Mother, KAL Grandmother, DIM Father
@ID:	bul|Bulgarian Language Corpus|SIM|1;07.23|female|||Target_Child|||
@ID:	bul|Bulgarian Language Corpus|PLA||female|||Mother|||
@ID:	bul|Bulgarian Language Corpus|KAL||female|||Grandmother|||
@ID:	bul|Bulgarian Language Corpus|DIM||male|||Father|||
@Date:	12-AUG-2020
@Media:	SIM_10723, audio, unlinked
@Situation:	at home
*KAL:	shte gi jadesh li?
%sit:	SIM za pyrvi pyt shte jade zhylti cheri domatcheta.
*SIM:	apche [: xapche], apche [: xapche], apche [: xapche], apche [: xapche].
*PLA:	ne e xapche.
*KAL:	zapovjadaj, izmiti sa.
*KAL:	izmiti sa.
*SIM:	nee!
*SIM:	apche [: xapche].
*KAL:	lapna go.
*SIM:	mamo.
*PLA:	kakvo e tova, mamo?
*SIM:	mama.
*PLA:	kakvo e?
*SIM:	apche [: xapche], apche [: xapche].
*PLA:	ne e xapche.
*PLA:	tova e cheri domat, mamo.
*SIM:	cheri.
%sit:	SIM xrani PLA i KAL s cherita.
*SIM:	baba.
*KAL:	zapovjadaj!
*SIM:	mama.
*SIM:	mama!
```

Fig. 1: A fragment of a transcript from the Bulgarian LabLing Corpus

Another important advantage of the CHAT-form of the transcripts is that, in view of the specific objectives that the researcher has in a given study, he or she can add lines of comments as needed. The comments can be of different nature: phonetic, morphological, situational, by the author, respectively, presented in the CHAT file as special lines: %Pho, %mor, % sit, %com, etc. For example, in the Bulgarian CHILDES-corpus additional %sit and %com lines are introduced, as well as the short presentation of deviations from the target language norm are given immediately before [: the norm unit]. Organised in this way, the speech resources prove to be very important as a reliable empirical basis for studying children's speech, as it is highly situational, abounds in deviations from the norm, and the values of the lexical deficit are too high. The following fragments of the Bulgarian CHAT-transcripts illustrate well these points:

TEF, 1;11:
*TEF: Nyama dam!
"No, I'm not giving it to you!"
%sit: TEF jumps on the bed and BAB is trying to catch her hand so that she won't fall. TEF keeps on jumping and puts her hands behind her back.
(2) ALE (1;1)
*ALE: Mama, mama!
"Mummy, mummy!"
%sit: She points to the door.
*VEL: Pri mama li iskaš?
"You want [to come] to mummy?"
%sit: VEL provokes the child by pretending not to understand the child's message.
*ALE: Mamo, mamo!
"Mummy, mummy!"
%sit: She implores with a whining voice.
*ALE: Mamma, mamma, maamma!
%sit: ALE pulls VEL rudely to the door.

The CHILDES provides the researchers with a package of specialised CLAN programmemes, which on their part can implement different types of analysis of the inserted dialogues. In that respect CLAN can automatically provide diverse statistical and substantial results out of the transcribed and coded data such as word frequency, lexical diversity and combinations, about a specific user's words and forms (for example, child language errors, such as specific deviations from the norm of the given language: the units of the so-called Baby Talk, onomatopoeia, super-generalisations, child and family occasions), etc. CLAN consists of two parts, namely the editor and the programmes. The second part of the CLAN provides the programmes for searching and analysis. The CLAN programmes which are of great interest with respect to interaction analysis include CHIP, COMBO, GEM, KWAL, and TIMEDUR. COMBO and KWAL allow users to search for all types of word and symbol combinations (MacWhinney and Wagner, 2010).

As an illustration to the aforementioned (see below) we will turn to the corpus of a Bulgarian girl – Alexandra (marked in the transcription with ALE) in order to demonstrate how conveniently and fast via FREQ programmeme from the CLAN set a frequent analysis could be implemented regarding the coded onomatopoeic elements in the main lines of the investigated child. After the start of CLAN, first we open the file (namely <probe.cha>).

Initial file: <probe.cha>

```
@Begin
@Participants:    ALE Alexandra Target_Child, VEL Velka Mother
@Birth of ALE:    29-JAN-1989
@Date:      27-MAR-1990
@Filename:        probe.cha
@Age of ALE:      1;01.29
@Situation:       at home
*ALE: Pyche [:pypche].
%sit:       poglezhda pypcheto si
*VEL:       Kyde e guceto grux-grux?
*ALE:       Gux-gux@o.
*VEL:       Grux-grux?
*VEL:       A kucheto, mamo, kyde e?
*VEL:       Kuche-e!
*VEL:       Au, njama go kucheto!
*VEL:       Kyde e kucheto?
*VEL:       am?
*ALE:       Bau-bau@o!
*VEL:       Vizh kakvo dyrzhi tati?
%sit:       pokazva kartinka s momiche, dyrzhashto kuchence.
*ALE:       Bau-bau@o.
@End
```

After that we press the command icon Commands, in which the necessary formula is typed (namely – freq +t*ALE +k +d*@o* +f probe.cha). Then we activate the operation and if the control system does not find any errors in the document structure, a new file is created (in particular – <probe.fr0.cex>), containing a list of child utterances with onomatopoeic elements (the trajectories of use of which, ordered alphabetically, are signed in the original *.cha file) and a quantitative analysis of the frequency of the coded elements.

```
Final file 1 (OUTPUT 2): <probe.fr0.cex>
freq +t*ALE +k +d*@o* +f probe.cha
Fri May 27 04:24:20 2005
freq (13-Apr-2001) is conducting analyses on:
ONLY speaker main tiers matching: *ALE
***************************************
From file <probe.cha> to file <probe.fr0.cex>
2 Bau-bau@o      : 19,22
1 Gux-gux@o      : 12
1 Pyche          : 9
------------------------------
3  Total number of different word types used
4  Total number of words (tokens)
0.750  Type/Token ratio
```

Thanks to the resources of the programme package CLAN, from files *.cha at the exit, it is possible to obtain different types of files. They could give information, which is necessary not only for statistical, but also for meaningful analysis of the corresponding chunk of speech. Particularly useful in this respect is the possibility (which is given by the KWAL command) to obtain an exit file (see below <probe.kw0.cex>) with isolated rows containing the element the researcher is interested in with the exact trajectories marked. In this way, the command Go in the source text can interpret the conversion context immediately.

```
Final file 2 (OUTPUT 2): <probe.kw0.cex>
kwal +t*ALE +k +s*@o* +f probe.cha
Wed Jun 01 10:21:24 2005
kwal (13-Apr-2001) is conducting analyses on:
 ONLY speaker main tiers matching: *ALE
***************************************
From file <probe.cha> to file <probe.kw0.cex>
----------------------------------------
*** File "probe.cha": line 12. Keyword: @o
*ALE:     Gux-gux @o.
```

```
----------------------------------------
*** File "probe.cha": line 19. Keyword: @o
*ALE:      Bau-bau @o!
----------------------------------------
*** File "probe.cha": line 22. Keyword: @o
*ALE:      Bau-bau @o.
```

Another advantage of the programme is that the main lines of the investigated child can be isolated not only independently but also in the context of one or several preceding or following lines, which is of particular importance for the research of speech ontogenesis. For example, from the initial demonstration file <probe.cha> we can receive an exit file containing information about the context of the child's utterance (in this case it is specified as a necessary line preceding the child's speech, which is encoded in the exit file with [– W1], resulting in the <probe.kw1.cex> file.

```
Final file 3 (OUTPUT 3): <probe.kw1.cex>
kwal +t*ALE +k +s*@o* -w1 +f probe.cha
Sun Jun 05 10:53:29 2005
kwal (13-Apr-2001) is conducting analyses on:
  ONLY speaker main tiers matching: *ALE
****************************************
From file <probe.cha> to file <probe.kw1.cex>
----------------------------------------
*** File "probe.cha": line 12. Keyword: @o
*VEL:      Kyde e guceto grux-grux?
*ALE:      Gux-gux @o.
----------------------------------------
*** File "probe.cha": line 19. Keyword: @o
*VEL:      Tam ?
*ALE:      Bau-bau @o!
----------------------------------------
*** File «probe.cha»: line 22. Keyword: @o
*VEL:      Vizh kakvo dyrzhi tati?
*ALE:      Bau-bau @o.
```

The contextual presentation of the analysed linguistic phenomena plays a vital role in the study of language categories, which are associated with sophisticated semantic complexes, as the preliminary treatment of the corpus prevents potential problems caused by polysemy and homonymy.

5. Conclusion

The sample demonstrations presented here do not exhaust all the advantages of the CHILDES system in the study of linguistic ontogenesis. The easy and user-friendly procedure for quantitative analysis, as well as the fact that CLAN is constantly improving, characterise it as a dynamic, efficient and convenient programme for working with large speech databases. This is what determines the widespread use of CLAN resources in the processing of the empirical material underlying modern models of linguistic ontogenesis.

The importance of CHILDES corpora and CLAN package of computer programmes can be summarised in the following potential applications:

Explanation in parent-child conversation using the CHILDES database;

- Modelling of input language system;
- Adequate and economical presentation of data related to highly deviant spontaneous child speech;
- Concise yet sufficient presentation of extralinguistic information, needed for the understanding of children's utterances;
- Automatic processing of speech databases;
- Automatic quantitative and statistical data analysis;
- Solving problems arising from homonymy, polysemy and other linguistic phenomena;
- Linguistic analysis at different levels;
- Multi-modality interface which allows for repeated use (see Popov and Popova, 2015).

In conclusion, it should be noted that the publication of the Bulgarian LabLing Corpus in the CHILDES system leads to an expansion of cross-linguistic research by adding another Slavic language to the database. In addition, the Bulgarian linguistic tradition acquires another universal easy-to-use standard for studying linguistic ontogenesis, thanks to which scientists will have the opportunity to quickly, accurately and reliably make comparisons among a large number of languages and build adequate typologies and sound modern theories.

6. Acknowledgements

This research was partially funded by the Bulgarian National Interdisciplinary Research e-Infrastructure for Resources and Technologies in favour of the Bulgarian Language and Cultural Heritage, part of the EU infrastructures CLARIN and DARIAH – CLaDA-BG, Grant number DO1-272/16.12.2019.

References

MacWhinney, B. (2000). *The CHILDES Project. Tools for Analyzing Talk. Vol. II, The Database*. Hillsdale: Lawrence Erlbaum Associates.

MacWhinney, B. and Wagner, J. (2010). Transcribing, Searching and Data Sharing: The CLAN Software and the TalkBank Data Repository. Gesprächsforschung. *Online-Zeitschrift zur verbalen Interaktion,* 11, 154–173. Available at: www.gespraechsforschung-ozs.de.

Bittner, D. et al. (2005). Aspect Before Tense in the Acquisition of Russian, Bulgarian, and German. In: *Text Processing and cognitive Technologies. V. 2. Proceedings of the VIII-th International Conference Cognitive Modeling in Linguistics (CML 2005)*. Moscow: MISA, Ucheba, 263–272.

Popov, D. and Popova, V. (2015). Multimodal Presentation of Bulgarian Child Language. In: *Proceedings of the 17th International Conference Speech and Computer (SPECOM 2015).* Springer International Publishing Switzerland, 293–300. Available at: https://www.springer.com/gp/book/9783319231310.

Kühnast, M. et al. (2004). Erwerb der Aspektmarkierung im Bulgarischen. *ZAS-Paper in Linguistics (Studies on the development of grammar in German, Russian and Bulgarian),* 33, 63–87. Available at: https://d-nb.info/1054690154/34.

Popova, V. (2006). *Child Language Early Grammar. Cognitive Aspects of Verbal Ontogenesis.* Veliko Tarnovo: Faber. (In Bulgarian)

Popova, V. (2016). *Event Modality. Early Ontogenesis*. Shumen: Konstantin Preslavsky University Publishing House. (In Bulgarian)

https://doi.org/10.33919/JCAL.23.1.4

TRANSLATION OF METAPHORS IN OFFICIAL AND AUTOMATIC SUBTITLING AND MT EVALUATION

Maral Shintemirova

University of Málaga, New Bulgarian University

maral.shintemirova@gmail.com

Abstract

One of the main aims of this work is to compare and analyse the translation of metaphors in subtitles as performed by human translators and by machine translation, and conduct MT evaluation.

The work considers two YouTube videos of a *Cyberpunk 2077* (2020) videogame walkthrough. The first video is in the original language (English) with English subtitles and the second one is an officially translated video in Russian, with Russian subtitles. Both videos have the same content, but in different languages.

Metaphors were extracted manually from selected audiovisual material in English by the usage of MIPVU (Metaphor Identification Procedure Vrije Universiteit). In order to achieve our aims, first the translation of these metaphors in the official Russian subtitles were analysed; secondly, their automatic translation into Russian as it appears on YouTube by Google Translate were analysed as well; after that the results were compared to find the similarities and the differences between the automatically translated version of the metaphors on YouTube and the translated metaphors in the official subtitling. Another aim is to perform Machine Translation (MT) evaluation using the BLEU (Bilingual Evaluation Understudy) algorithm and to determine the errors made by MT while translating metaphors in the analysed subtitles.

Three examples, which were taken from the videos, are presented in the format of cases. The cases show different metaphors and the situations they were used in and analyse why these metaphors were used in that particular situation, how metaphors were identified there, how they were translated and why they were translated exactly in this way. Furthermore, the machine translation of the same metaphors is analysed and a comparison between them is made. The topic of the speech recognition process and the metaphor identification procedure is also touched upon.

The results demonstrate that although machine translation is able to translate frequently used, popular metaphors, or metaphors, the literal translation of which retains the meaning, it is still difficult for the machine to recognise original author's metaphors or to translate using the context of the situation. The results could encourage training the machine to recognize metaphors and to create a larger database of metaphors to identify them.

Keywords: *metaphor, machine translation, MT evaluation*

Introduction

The relevance of the topic and research problems

In the era of globalisation, automatic systems do not stand still. The main task for machine translation technologies is to ensure accurate translation, depending on the context, as well as application of technologies to particular areas.

A tremendous amount of content has started to appear in English, a lot of which is audiovisual content. Nowadays, visual materials have gained certain popularity and continue to do so. Watching movies, TV shows, series and videos has become a significant source of entertainment in recent years. According to the Statista website, over three billion internet users watched at least one streamed or downloaded video every month in 2020 (Statista.com). This indicates a vast audience that very often consumes various audiovisual content.

When it comes to the audiovisualisation of communication in today's technologically driven multimedia culture, the value of moving images, complemented by sound and text, is critical. We are surrounded by screens in both our professional and personal lives, as they are a frequent element of our socio-cultural milieu. We spend a lot of time in front of screens – at home, at work, in public places, in libraries, cafes, restaurants and cinemas – and we consume a lot of audiovisual products in order to be entertained, to get information, to do our jobs, to learn and develop, and improve our professional careers. The abundance of moving images and their impact on our lives illustrate the audiovisualisation of communication in our time and age (Díaz Cintas and Remael, 2013).

Video materials are included in the field of Audiovisual Translation (AVT). AVT is defined as "…all translations – or multi-semiotic transmission – for production in any medium or format, as well as new areas of media accessibility" (Díaz Cintas, Orero, Remael, 2007). Several perspectives on the translation of video materials lead to the main goal of AVT: to create a translation that respects the cultural identity of the source language, while remaining accessible to other target audiences. In order to achieve this goal, it is necessary to use different translation procedures for audiovisual materials. There are about ten different ways of translating audiovisual materials, yet there are "three main ones: dubbing, subtitling and voice-over" (Díaz Cintas, Remael, 2007). This paper will concentrate on subtitling as one of the most popular and frequently used ways of translating audiovisual content. Díaz Cintas and Remael's definition of subtitling is as follows:

> Subtitling may be defined as a translation practice that consists of presenting a written text, generally on the lower part of the screen, that endeavours to recount the original dialogue of the speakers, as well as the discursive elements that appear in the image (letters, inserts, graffiti, inscriptions, placards, and the like), and the information that is contained on the soundtrack (songs, voices off).
>
> (Díaz Cintas, Remael, 2007).

When mentioning audiovisual materials, it is assumed that they can be accessible to everyone. To do this, they must be available in different languages so that people who speak other languages can also understand the content. In this case, translators act as bridges that help people comprehend the material. In order to facilitate and speed up the work of translators, machine translation is sometimes used. Machine translation (MT) of natural languages, which was initially proposed in the seventeenth century, is now a reality (Hutchins, 1995). Computer programmes generate translations –

not perfect translations, because sometimes not even human translators can achieve that goal. As Hutchins (1995) explains,

> it is quite clear from recent developments that what the professional translators need are tools to assist them: provide access to dictionaries and terminological databanks, multilingual word processing, management of glossaries and terminology resources, input and output communication.

MT faces the global task of translating content into various languages for a wide audience, while maintaining not only the meaning and the style, but also the emotional overtones. With the development of translation as a discipline and its rotation to a 'cultural turn' in the 1990s, the operational unit of translation was not a word or a text, but a whole culture (Bassnett, 1997). From that moment on, the object of translation was the text integrated into the network of relations between the source and translating cultural signs (Bassnett, 1997). After the advent of the 'cultural turn,' linguists were faced with the task of creating a system that recognises the cultural context, turns of speech, and language figures.

The translation of subtitles is not an easy task for translation programmes, therefore, when programme generates a script, a hybrid approach is often used, where, after the translation program has translated the audiovisual material, the human translator is in charge of editing and post-processing. Subtitles must appear in sync with the image and the original dialogue, provide a semantically acceptable explanation of the source language dialogue, and remain visible on the screen for viewers to read (Díaz Cintas, Remael, 2007) and these are tasks that MT cannot yet do. Also, in terms of space, screen sizes are limited, and the target text will need to adjust to fit the screen width. This means that the subtitle will be between 32 and 41 characters per line in a maximum of two lines (Díaz Cintas, Remael, 2007). However, some programs do this automatically themselves, such as YouTube (https://support.google.com/youtube/answer/7296221?hl=en).

The translation of subtitles cannot be verbatim, as literal translation almost always results in a poor translation. The audiovisual content and the subtitles should fully harmonise with each other for a high-quality translation, but no MT engine relies on visual content (Díaz Cintas, Remael, 2007).

Translators take into account the features of speech transmission, style, emotions, terminology, dialects, and even gender and age, while MT cannot determine these criteria. One of the most challenging tasks is adapting or reflecting culture, because the translation must be understandable for the people whose culture differs from the culture in the source material. If the translator is faced with the task of leaving the cultural aspects as they were in the original and bringing the viewer closer to the audiovisual content, then MT cannot take into account the aspects of speech turnover.

Some of the representatives of culture are figures of speech, such as puns, irony, satire, euphemism and others (Regmi, 2015). The difficulty lies in the fact that they are culturally coloured, and that becomes an even more significant challenge for the translator. One of the inevitable difficulties of AVT is the presence of metaphorical utterances in the source text. James Dickins defines the metaphor as follows:

> 'Metaphor' is defined [...], as a figure of speech in which a word or phrase is used in a non-basic sense, this non-basic sense suggesting a likeness or analogy [...] with another more basic sense of the same word or phrase.
>
> (Dickins, 2005)

Many works have been written about metaphors and their translation: "The Translation of Metaphor" by Peter Newmark (1980), "Biblical Metaphors and Their Translation" by Jan De Waard (1974), "Metaphor and Translation" by Richard Trim and Dorota Liwa (2019), etc. The problem of translating metaphors is one of the most complex and essential ones since a metaphor is the embodiment of original, emotionally coloured images that perform one of the most critical tasks in the text and audiovisual materials – influencing the reader's or viewer's imagination.

There are two non-verbal channels in an audiovisual text: an auditory one and a non-verbal visual one, which encompasses everything people see on screen, in addition to the verbal channels of discourse, spoken language, and written language. When a metaphor is identified in subtitles and it does not have an analogous expression in the target language, these complications become much more severe (Pedersen, 2015).

Research question

This work will focuses on MT evaluation of metaphors in official subtitles and in MT, on the comparison between the translation performed by humans and that provided by a MT engine. Two videos from the YouTube web service with the videogame walkthrough are taken into consideration. The name of the videogame is *Cyberpunk 2077*. The content of both videos is identical; however, they are in different languages – the original video is in English and officially translated video is in Russian. In these videos subtitles are displayed on the screen. Metaphors are identified manually from in the English subtitles. After that, the official Russian translation of those metaphors in subtitles are analysed, as well as the same metaphors automatically translated into Russian in YouTube by Google Translate. The results are compared and evaluated by means of the BLEU (bilingual evaluation understudy) algorithm in order to determine the errors made by MT while translating metaphors in subtitles.

The process of identifying metaphors can be a difficult task. In such cases, technology could help. However, as Saldanha (2009) argues, "a number of metaphor retrieval computer tools have been developed, but they have not made an impact in the field, partly because they are not widely available and partly because their performance is still not particularly high." Despite the fact that technologies are developing rapidly, not all areas use them yet. This indicates a lack of attention to this problem, which nevertheless exists. In this regard, as already mentioned above, all metaphors for this research were identified and assembled manually.

This study aims to find out how the machine translates metaphors in subtitles, whether the machine translation of official subtitles differs significantly from human translation, whether it is possible to use the MT of metaphors without a human translator editing them for further usage, if there is much to strive for in the refinement of MT of metaphors.

The subject of the study are two videos from the YouTube web service of the *Cyberpunk 2077* videogame walkthrough from PaNiKeR player's YouTube channel in the original language (English). The source language of the material is English, the target is Russian.

Aims and tasks

The aims of this study are to analyse the official translation and the MT of metaphors in subtitles, to compare them, and to determine the errors made by MT while translating those metaphors.

To achieve these goals, the study solves the following **tasks**:

1.) considering both metaphor and metaphorisation processes;

2.) identifying metaphors by means of the MIPVU (Metaphor Identification Procedure Vrije Universiteit);

3.) studying two videos of the *Cyberpunk 2077* videogame walkthrough on YouTube;

4.) analysing the official translation of metaphors;

5.) analysing MT of metaphors in subtitles;

6.) comparing and analysing the two translations;

7.) evaluating the results as provided by the BLEU algorithm (bilingual evaluation understudy).

The length of the video material is about 9 hours. The total number of metaphors is 269.

Methodology

When gathering and analysing the material, the following **research methods** are used: general approach, discourse analysis, stylistic analysis, definition analysis, component analysis. Both primary and secondary data will be used in this work. Moreover, we apply the product-oriented research methodology which investigates the translation product, in my case subtitles in target language, from different perspectives. It also includes critical discourse analysis, which can be also applied by means of a quantitative or a qualitative method (Saldanha and O'Brien, 2013). Both quantitative and qualitative methods are used in this study, as we describe the processes of identification and translation of metaphors, as well as provide figures using the BLEU algorithm for the evaluation of MT.

As for the research aims, we use basic research aimed at developing knowledge, as well as applied research aimed at analysing how translation tools translate metaphors.

For **data research,** we use the qualitative type, which involves gathering and analysing non-numerical data, such as videos, to understand concepts, opinions, or experiences. It is used to gather in-depth insights into a problem and generate new ideas for research. We collect existing data in the form of texts and phrases for this study and future projects.

These works build the foundation of the paper, providing the basic concepts of AVT, translation of metaphors, features of the translation of subtitles and, separately, the features of the translation of metaphors, as well as the general study of MT and MT of metaphors. They help to understand the functioning of the software, the web services and the specifics of the translation of metaphors.

The results of this study can be used by specialists in the field of translation and linguistics for further research, to assist in the translation of metaphors in subtitles, as material for analysis in educational institutions or in practical English language teaching, as well as for studying metaphors for general or specific purposes. In addition, this study can help in the further study and improvement of both MT in general and MT of metaphors specifically. It can also help in improving the evaluation system or developing the tools for evaluating metaphors, and may contribute to the development of technologies that will help identify the metaphor in subtitles, and help in translation.

The process of identifying and translating metaphors

Metaphor identification procedure

The study of metaphor in recent decades has become one of the most widespread trends in linguistics. Despite the fact that these studies were based on the correlation of language, thinking, and the modelling of knowledge about the world through metaphor, it became a problem to identify metaphors. Attempts to solve this problem have led to the creation of various metaphor identification procedures, which differ both on theoretical grounds and procedurally (Mishlanova, Suvorova, 2017).

Metaphors in the *Cyberpunk 2077* videogame

Cyberpunk 2077 is an action-adventure computer game in the Open world (virtual world), developed and released by the Polish studio CD Projekt, in which players are free to explore and achieve their goals. The genre is cyberpunk. The game is partly interactive, *i.e.* players choose one of the suggested actions or phrases. Each decision taken leads to different storylines and game endings. The plot takes place in 2077 in Night City, a fictional North American city from the Cyberpunk universe (cyberpunk.net). Gamers control a customisable protagonist named V, whose gender can be chosen. The character works as a mercenary and has hacking and combat skills. The game was released on December 10, 2020 on PlayStation 4, Stadia, Windows and Xbox One[1].

The original video game is in English and it was the basis for all versions of the localisation into other languages. The game has been translated into 18 languages, 10 of which are dubbed. The Russian translation was based on the original version in English. The translation process lasted a year and a half and the company considers it one of the most ambitious translation projects in gaming industry (en.cdprojectred.com).

The Russian version of the game is quite aggressive, the translators inserted foul language even if there were no such phrases in the original version, and added slang to intensify the atmosphere of the game. By that, players may experience the environment of the characters' and city's decadence, the style of street life and gang warfare, where every population group swears differently.

There are some examples in which metaphors are identified and their official translation in the subtitles to the video is analysed. Metaphors are

[1] For more detailed information – https://en.cdprojektred.com/; cyberpunk.net.

defined manually while watching the videos and reading subtitles. They are analysed following the steps of MIPVU procedure.

Case. At timecode: 9.57 V is in a car with Sebastian Padre Ibarra, who can 'fix' any problem. People come to him, asking for help to rob, hack a system, eliminate a person. He gives the order to a contractor who performs the task. V is having a conversation with him about life. V was born in Heywood, went to live in Atlanta for several years, but returned to his hometown.

> Padre: "You know Heywood. It has strong roots – ever watered by the same blood[2]."

The metaphors are *'it (city) has strong roots'* and *'(city) ever watered by the same blood.'* The lexical units of this sentence are the following: 'it has strong roots,' 'it ever watered by the same blood,' 'same blood.' We consider 'it has strong roots' and 'it ever watered by the same blood' units. Both phrases are related, the second phrase complements the first.

'It (city) has strong roots:'

a) **Contextual meaning**: As the roots are nourished or watered, they become strong. The city is made stronger by the events, taking place in it.

b) **Modern basic meaning**: Strong (adj.) – powerful; having or using great force or control; root (n.) – the part of a plant that goes down into the earth to get water and food and holds the planet firm in the ground; roots (n.) – family origins, or the particular place you come from and the experiences you have had living there[3]. In a direct sense, it means that the plant has strong roots,stands firmly in the ground, and grows for a long time unshakably.

c) **Contrast and similarity**: A city cannot have roots, but it, like roots, can become stronger.

This is a 'stock metaphor,' *i.e.* an ordinary metaphor with an aesthetic function, and there may be equivalents in translation.

'(City) ever watered by the same blood.'

a) **Contextual meaning**: The city is in turmoil, with people dying because of gangs or newcomers. 'Blood' means the blood of dead people, and 'same' means the deaths of local people who cannot defend themselves. Murders and robberies make the city's gangs stronger.

b) **Modern basic meaning**: Ever (adv.) – at any time; watered (v., passive v.) – to pour water onto plants or the soil that they are growing in; same (adj.) – exactly like another or each other; blood (n.) – the red liquid that is sent

[2] https://www.youtube.com/watch?v=TTQ1L5qpVwM&t=1294s – at timecode: 9.57.

[3] dictionary.cambridge.org.

around the body by the heart, and carries oxygen and important substances to organs and tissue, and removes waste products[4].

c) **Contrast and similarity**: The phrase has a direct meaning of the word 'blood,' but in this case there is a metaphorisation – just as water nourishes the roots of a plant, so the deaths of people make the city and gangs stronger.

This statement is metaphorical and is of 'original type' expressing the creator's idea.

Analysis of the official Russian translation of metaphors in subtitles

Regarding the decomposition of the translation process into certain stages and procedures, there is no single view, just as there is no single definition of the terms 'strategy' and 'translation technique'. Each translator can have their own translation strategy, allowing them to decide what is less important and can be omitted in a particular translation situation, or how to conduct a translation (Garbovskii, 2007). Therefore, we consider the provided translation of the game in the video, trying to understand how the translator translated the metaphor and what guided them during the process. We take the stages of metaphor translation proposed by Newmark as a basis while we analyse metaphors translation from the video.

When analysing the translation in subtitles, it is important to take into account the fact that depending on the genre of the game, the number of dialogues may be different, respectively, and the length of the subtitles may vary as well. Since subtitles can act as a shortened version of the original, they look concise and compact on the screen in order to complement the action of the game, and not to draw attention to themselves (Chandler, 2006). Therefore, in some unimportant moments, the translator may omit a metaphor or shorten it, thus it will look different in translation.

We consider the translation into Russian, presented in the Russian subtitles of the video, of the same cases that were discussed above.

Case.

> Padre: "You know Heywood. It has strong roots – ever watered by the same blood."
>
> Падре: «Это же Хейвуд. У него корни крепкие. И кровь их питает все та же».

The metaphor '*it (city) has strong roots*' was translated as «у него (города) корни крепкие» (it (city) has strong roots). Being 'a stock' metaphor, it describes an abstract concept that has an emotional impact on the viewer. In Russian, there is a metaphor «иметь сильные корни» (to have strong roots),

[4] dictionary.cambridge.org.

which means to be in good relationships with the family, to be strong because one's connection to the family is strong. The metaphor in the video refers to the city, but it is strong not because of good conditions, but because of the blood that 'fuels' it and because of the gangs that run the city. The phrase was translated verbatim, but it remains metaphorical; the viewer understands the message. The first and second metaphors of this case are related, and although the phrases have opposite meaning in English and Russian, the viewer senses the negative meaning from the context – blood makes the city strong. The translator used 1 and 2 translation techniques, a metaphorical image was left and a Russian equivalent was chosen.

The second metaphor *'(city) ever watered by the same blood'* was translated as «кровь их питает все та же» (the same blood still feeds them). Grammatically, the sentence was changed a bit, the original is in the passive voice, the translation is in the active. As this is 'an original' metaphor, the translators had to use creativity to choose the right translation. There is no such metaphor or equivalent in Russian. The metaphor is neither paraphrased nor has an extra explanation, but it was translated while retaining the metaphorical image. The player and the viewer understand the meaning from the context and in conjunction with the first metaphor.

Results

Translating video game subtitles is a difficult and at the same time creative task. The translator must take into account a large number of factors before starting to translate. It has to be taken into consideration that the translation of video games differs from the translation of books to a greater extent by the genre and style of conversation, and from films by the size of subtitles and the purpose of subtitles. If subtitles complement the plot in the film, helping to delve into it, then in the game they can distract, so it is necessary to think about the way of translation and the size of the lines.

Numerous elements found in the text are localised and translated: game names, proper names, invented words and expressions, metaphors or units that do not have counterparts in other languages, the translation of which in some cases requires a special skill and creativity. One of the main goals of translation of videogames is the linguistic and cultural adaptation of video games, or the transfer of cultural information in the process of translation.

The author's 'original' metaphors are the most difficult to translate, since in most cases there is no equivalent in TL and the creators invented them themselves. Out of 269 metaphors in *Cyberpunk 2077*, more than 150 were 'original' metaphors and the translators, not finding a suitable equivalent, translated some of them as they considered correct, replacing them with phrases close in meaning or removing them, while preserving the style and meaning of the utterance. Since the dialogues were in an informal street style, there was a lot of obscene metaphorical vocabulary, and they were intertwined. The translators managed to translate mostly verbatim, while maintaining metaphorical images.

Since there is no general technique for defining metaphors, it can be a complex task to distinguish whether a phrase is metaphorical or not. Sometimes a metaphor becomes an integral part of a conversation and is not noticeable. In other cases, the authors may add new, previously unseen original metaphors to works of art. The situation is complicated by the fact that there is also no general technique for translating metaphors, there are only recommendations.

Translators have to rely on their own guesses, intuition and experience and try to find suitable equivalents. If this is not possible, then they either have to explain the metaphor, which eliminates the aesthetics and ruins the style, or omit the metaphor, which can also harm the style and possibly make the meaning of the sentence incomprehensible. Translators take the decision to save or remove a metaphor based on what type of subtitles they work with, the number of individually authored metaphors in the text (whether the text is overloaded) and how appropriate it will be in a particular situation to resort to metaphorisation at all. Therefore, translators have to decide whether to preserve a metaphor, reproduce the corresponding construction in translation or omit it, compensating in some other, no less expressive way. However, in videogame subtitles, the stylistic effect is as important as the idea of the plot itself. Sometimes translators do not take risks and translate verbatim, referring to the viewers' understanding of the context. Therefore, the translation of metaphors is a real art.

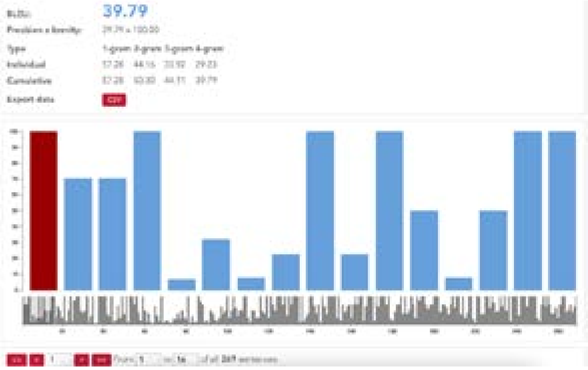
MT analysis and MT evaluation of cases

Analysis of MT of metaphors in the *Cyberpunk 2077* YouTube video

As previously stated, automatic YouTube subtitles appear on the screen after automatic speech recognition (ASR). After speech has been recognised, it is translated. YouTube creates a translation of subtitles using Google Translate. However, as mentioned earlier, the translation algorithm of Google Translate is confidential information.

Machines are trained to recognise expressions, and many of them are already embedded there, so it is not difficult for the machine to translate common metaphors. However, difficulties may arise when translating adapted, original and author's metaphors, since they are unique. As was mentioned earlier, we considered MT using the example of a video of the *Cyberpunk 2077* game walkthrough in English with automatic subtitles in Russian enabled.

After watching the videos and making a list of metaphors and a list of officially translated metaphors in subtitles, we proceeded to create a list of the automatically translated metaphors on YouTube, which was performed manually as well. Having manually created separate files with the original text in English, the official translation in Russian and MT in Russian, we uploaded them to the website and the calculation was performed. In the initial stage, we took 269 metaphors. The human translation is considered to be 100.00, and the calculation formula described in the previous paragraphs is used to derive the result of MT evaluation – 39.79 BLEU score. Each column in the following picture describes a different metaphor: the higher the column, the higher the similarity between human translation and MT. **Picture 1.** represents overall results of the analysis of 269 metaphors.



BLEU: 39.79
Precision x brevity: 39.79 x 100.00
Type: 1-gram 2-gram 3-gram 4-gram
Individual: 57.28 44.15 33.92 29.23
Cumulative: 57.28 50.30 44.11 39.79
Export data:

Picture 1. BLEU score of 269 metaphors[5]

In order to compare the translation of subtitles made by humans and the MT of automatic subtitles on YouTube, we performed an analysis using the BLEU metric of the same examples from the cases which were examined in the previous section.

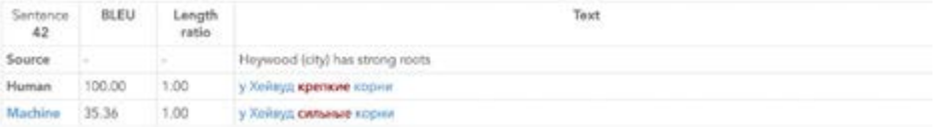
Case.

> Padre: "You know Heywood. It has strong roots – ever watered by the same blood."

Metaphor '*it (city) has strong roots*': the official Russian subtitles (OS) are «у Хейвуд крепкие корни» (Heywood has sturdy roots), while the MT of subtitles is «У Хейвуд сильные корни» (Heywood has strong roots) has

[5] https://www.letsmt.eu/Bleu.aspx.

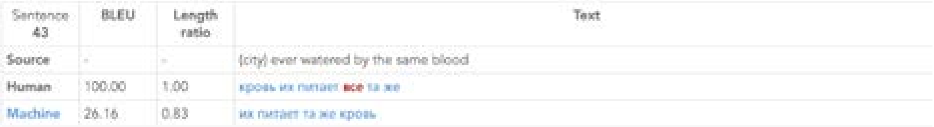
35.36 score. The number of words matches (four words), the name of the city (Хейвуд) and the verb tense (Present Simple) are translated in the same way, plural number, sequence of words (preposition + subject noun + adjective + object noun) are also the same. However, there are lexical inconsistencies: «крепкие» (OS) (sturdy) and «сильные» (MT) (strong). Although they have close meaning, in Russian they are some differences. «Крепкий»[6] (OS) means "durable", such as when referring to something that is difficult to break or tear; «сильный»[7] (MT) means possessing great physical strength, being powerful. That is, while «Крепкий» (OS) means that something is sturdy and hard to break, «сильный» (MT) means that this object/person can break something. If we talk about a person, they can be sturdy (крепкий), but it does not mean that they are strong (сильный). Despite the difference, the metaphors persist, since these metaphors are present in both languages.



Sentence 42	BLEU	Length ratio	Text
Source	-	-	Heywood (city) has strong roots
Human	100.00	1.00	у Хейвуд крепкие корни
Machine	35.36	1.00	у Хейвуд сильные корни

Table 1. BLEU score of 'it (city) has strong roots' metaphor.

The second metaphor appearing in this example, *'(city) ever watered by the same blood'* is «кровь их питает все та же» (the same blood still feeds them) in the OS and «их питает та же кровь» (they are nourished by the same blood) in the MT. The metaphor is preserved in both versions, and the meaning is identical, but BLEU showed only a 26.16 score. The similarities are the verb tense (Present Tense) and the meaning. The differences are the following: number of words (six in the OS and five in the MT); a word «все» (still) added in the OS, that reinforces the meaning that this is still happening; the OS is in the active voice, the MT is in the passive voice; depending on the active/passive voice, the structure and the sequence of the words have changed.



Sentence 43	BLEU	Length ratio	Text
Source	-	-	(city) ever watered by the same blood
Human	100.00	1.00	кровь их питает все та же
Machine	26.16	0.83	их питает та же кровь

Table 2. BLEU score of '(city) ever watered by the same blood' metaphor.

[6] Ozhegov dictionary (ТекстоЛогия) – https://ozhegov.textologia.ru/definit/krepkiy/?q=742&n=176804.

[7] Ozhegov dictionary (ТекстоЛогия) – https://ozhegov.textologia.ru/definit/silniy/?q=742&n=203153.

Results

As stated above, MT can significantly save time and, in general, modern neural translation systems perfectly cope with the task of translating the general meaning of a message. However, they have not yet reached the level of perfection when human participation is completely excluded from the translation process.

The biggest problems with the results of MT are related to semantics, since the translation of semantic constructions requires databases that are not currently developed. In addition, translations of complex grammatical, syntactic and lexical constructions cause considerable difficulties. Further development of automatic translation is associated with the possibility of a holistic assessment of texts performed using computer translation systems. It is an adequate and complete assessment that will help to identify and systematise all the shortcomings of the programs so that these problems can be solved in the future.

The most popular metric when analysing the quality of MT is BLEU, but like any metric, it is only close to human translation, but not identical.

According to the results gained after calculation by the BLEU algorithm on the website, MT coincided by 39.79 points with human translation, which is quite a high indicator, although far from perfect. Out of the 269 metaphors found in the videos, 111 showed a result of more than 50%, of which 57 showed a 100% match. Although 158 metaphors showed results below 50%, 81 of them were even below 10.00 score. According to such data, it can be concluded that MT translation does not cope very well with metaphors, especially with author's ones, and needs further refinement and training.

Translating metaphors is not an easy task for a human translator. Besides the fact that the metaphor needs to be defined, it needs to be translated correctly. MT often makes mistakes and is not very good at translating fiction or text in which there is live or recorded everyday speech, where there are no clichés, but rather a lot of artistic turns and figures of speech. Metaphors in such texts are sometimes created by individuals and they may not even be immediately identified, therefore, the machine cannot always recognise and translate them. The question remains, is it worth leaving such a complex and creative work as translating metaphors to a machine, or should a person do it anyway?

References

Baker, M. and Saldanha, G., (eds.). (2009). *Routledge Encyclopedia of Translation Studies.* 2nd ed. London: Routledge.

Bassnett, S. (1997). Moving Across Cultures: Translation as Intercultural Transfer. In: J. M. Santamaría et al., (eds.). *Trasvases culturales: Literatura, cine, traducción* (2). Universidad del País Vasco, Facultad de Filología, Departamento de Filología Inglesa y Alemana, 7–20.

Chandler, H. M. (2006). *Game Production Handbook.* 1st ed. Hingham Mass: Charles River Media, 420.

Díaz-Cintas, J. (2009). Audiovisual Translation: An Overview of its Potential. In: *New Trends in Audiovisual Translation*, 1–20. Briston: Multilingual Matters

Díaz-Cintas, J., Orero, P., Remael, A., (eds.). (2007). *Media for All: Subtitling for the Deaf, Audio Description, and Sign Language.* Amsterdam: Rodopi

Díaz-Cintas, J., Remael, A. (2014). *Audiovisual Translation: Subtitling*. London: Routledge.

Díaz-Cintas, J., Remael, A. (2020). *Subtitling: Concepts and Practices*. London: Routledge.

Garbovskii N. K. (2007) *Theory of Translation.* 2nd ed. Moscow: Moscow University Press, 542. (In Russian)

Hutchins, W. J. (1995). Machine Translation: A Brief History. In: Koerner, E. F. K. and Asher, R. E., (eds.). *Concise History of the Language Sciences: From the Sumerians to the Cognitivists*. Oxford: Pergamon Press, 431–445.

Hutchins, J. (2010). Machine Translation: A Concise History. *Journal of Translation Studies*, 13(1&2), 29–70. Available at: http://www.hutchinsweb.me.uk/CUHK-2006.pdf.

Mishlanova S. L., Suvorova M. V. (2017). Evaluation of Metaphor Identification Procedure VU (MIPVU) by the Criteria of a Truly Scientific Method. *Perm University Herald. Russian and Foreign Philology*, 9(1), 46–52. Available at: http://www.rfp.psu.ru/archive/2017.9.1/mishlanova_suvorova.pdf. (In Russian).

Neuman, Y. et al. (2013). Metaphor Identification in Large Texts Corpora. *PLoS ONE*, 8(4), e62343.

Newmark, P. (1980). The Translation of Metaphor. *Babel: International Journal of Translation*, 26(2), 93–100.

Newmark, P. (2008). *A Textbook of Translation*. 12e ed. Harlow: Pearson Education, 292.

Pedersen, J. (2015). On the Subtitling of Visualized Metaphors. *The Journal of Specialised Translation*, 23, 162–180.

Regmi, L. R. (2014). Analysis and Use of Figures of Speech. *Journal of NELTA Surkhet*, 4, 76–80.

Saldanha, G., O'Brien, S. (2013). *Research Methodologies in Translation Studies.* Manchester: St. Jerome Publishing.

Web sources:

BLEU evaluation website (https://www.letsmt.eu/Bleu.aspx).

CD PROJEKT (en.cdprojectred.com).

Cyberpunk 2077 (videogame cyberpunk.net).

Google Support YouTube (https://support.google.com/youtube/answer/7296221?hl=en).

Oxford for Learners Dictionary (oxfordlearnersdictionaries.com).

Cambridge Dictionary (https://dictionary.cambridge.org/).

Ozhegov Dictionary (https://ozhegov.textologia.ru/definit/krepkiy/?q= 742&n= 176804).

https://doi.org/10.33919/JCAL.23.1.5

GENDER-NEUTRAL LANGUAGE USE IN THE CONTEXT OF GENDER BIAS IN MACHINE TRANSLATION (A REVIEW OF LITERATURE)

Aida Kostikova

New Bulgarian University, Ghent University

aida.kostikova@ugent.be

Abstract

Gender bias has become one of the central issues analysed within natural language processing (NLP) research. A main concerns in this field relates to the fact that many NLP tools and automatic machine learning systems not only reflect, but also reinforce social disparities, including those related to gender, and language technology is one of the areas in which this issue is pronounced. This paper analyses the problem of gender-neutral language use from the standpoint of gender bias in machine translation (MT). We determine which types of harms can be caused by the failure to reflect gender-neutral language in translation, provide the general definition of gender bias in MT, describe its sources and provide an overview of existing mitigating strategies. One of the main contributions of this work is that it focuses not only on females, but also non-binary people, whose linguistic visibility has been receiving only limited attention from academia. This literature review provides a firm foundation for further research in this area aimed at addressing the problem of gender bias in machine translation, especially bias linked to representational harms.

Keywords: *gender bias, machine translation, NLP tools, gender-neutral language use, non-binary gender*

1. Introduction

As the adoption of gender-neutral language (GNL) becomes more widespread, it is increasingly important to consider how these trends can be reflected in natural language processing (NLP) applications, especially given the fact that the purpose of GNL is to "reduce gender stereotyping, promote social change and contribute to achieving gender equality" (Papadimoulis, 2018, 3). Failure to adopt more equitable and balanced linguistic practices can lead to bias associated with representational and, ultimately, allocational harms (Crawford, 2017). The major concerns raised by the researches in this field are related to the fact that any type of bias in technology can be detrimental for ensuring social justice, as by hindering the visibility of speech patterns of certain groups and allocating certain stereotypes to them, such systems can perpetuate inequality (Levesque, 2011; Régner et al., 2019).

While much of prior work in the field of gender bias studies gender identity, most is built on techniques which assume that gender is binary. At the same time, there is growing recognition of non-binary gender identities, with numerous ways to refer to non-binary people or to simply not indicate a binary gender (Sun et al., 2021). That is why in it is necessary to take into account strategies aimed at increasing the linguistic visibility of non-binary people in NLP, and, in particular, in machine translation (MT). In this paper, we attempt to analyze the problem of gender bias from the standpoint of GNL use. The goal is to define and classify types of gender bias generated by a biased MT, and identify harms which might occur due to the failure to reflect gender-neutral language in translation; in addition, we provide an overview of gender-neutral strategies and discuss a rationale for their use. Special attention is paid to non-binary language and its application in machine translation.

2. The issue of gender bias in languages/translation/MT

Although natural language processing (NLP) research does not directly involve human subjects (Hovy and Spruit, 2016; Bender et al., 2021), its engagement with language – the main mediator of the human experience –, which shapes communication as well as such cognitive processes as categorization and perception – raises the question of the social impact of language technologies. The major concern raised by researchers in this field is that bias in technologies can undermine any efforts to establish social justice and equality, as they have a direct impact on the allocation of resources integration and the inclusion of certain social groups (Hovy and Spruit, 2016). Among the narrower, but no less significant, issues related to bias in NLP and languages, are exclusion, stereotyping, bias reinforcement and denigration (Bender et al., 2021).

Overall, there is a close link between bias in technology and prejudice (Ferrer et al., 2021), which has certain psychological and sociological implications (Bourguignon et al., 2015). Machine translation (MT) systems are no exception, as they are known to reflect asymmetries, including those related to gender (Prates et al., 2020), and this phenomenon can be manifested in many ways, with issues ranging from gender stereotyping (Olson, 2018) to over-reliance on the so-called "masculine default" (Schiebinger, 2014). Particular attention must be paid to adverse effects that MT systems may have, as it is one of most widely used artificial Intelligence (AI) applications on the Internet, which is also employed indirectly, e.g., through social media (Monti, 2020).

2.1 Bias statement and implications of gender bias

Overall, a model can be regarded as biased in cases when, while being created by and for people (Schnoebelen, 2017), it "systematically and unfairly discriminates against certain individuals or groups in favor of others" (Friedman and Nissenbaum, 1996) and entails risks associated with social exclusion and stigmatisation (Bender et al., 2021). Bias can be represented in multiple parts of a system, including the training data, resources, pretrained models, and algorithms themselves (Zhao et al., 2018; Bolukbasi et al., 2016; Caliskan et al., 2017; Garg et al., 2018), which can lead to the production of biased predictions and the further reinforcement of biases present in the training sets (Zhao et al., 2017).

Such systems can, therefore, cause representational harms (i.e., diminishing the role and exclusion of social groups and their identity) or allocational harms (i.e., cases were a system limits the access to resources for certain groups or allocates them in an unfair way) (Crawford, 2017). By drawing on the classification used by Savoldi et al., we also consider such harmful dynamics within representational harms, as stereotyping and under-representation (Savoldi et al., 2021). Stereotyping involves the propagation of generalized beliefs about a social group, for example, by assigning less prestigious occupations or negative physical characteristics to women. Under-representation refers to the cases where the visibility of certain social groups is reduced, which in most cases affects women and non-binary individuals. More emphasis will be placed on the second category of harms (under-representation), as it involves cases of misgendering and ignoring gender-neutral forms, which is precisely the object of our study.

Within the classification framework developed by Dinan et al., who defines harms based on gender dimensions (bias when speaking about someone or gender of the topic, bias when speaking to someone or gender of the addressee, and bias from speaking as someone or gender of the speaker), failure to convey gender-fair language can be described as, on the one hand, misrepresentation when talking "about" certain groups, and on the other hand as reduced visibility of the language used "by" speakers of such groups, which can be detrimental for reflection of their identity and communicative repertoires. In other words, an MT system which does not recognize or reflect certain linguistic expressions of gender might present a barrier for communication and produce an output that "indexes unwanted gender identities and social meanings" (Dinan et al., 2020).

In a broader context, such trends also have an impact on indirect stakeholders, because a biased MT system does not only contribute to the reinforcement of stereotypical assumptions and prejudices (Levesque, 2011; Régner et al., 2019), but promotes language features used by the dominant group, and consequently their establishment as appropriate or prestigious variants (Tallon, 2019). The issue is compounded by prioritization of the overall quality of an MT output, which in most cases is viewed as acceptable by an MT user and perceived as the linguistic norm in a given language (Martindale and Carpuat, 2018). Therefore, there is a close link between representational and allocational harms, which manifests itself in performance disparities across users in the quality of service *(*Savoldi et al., 2021*)*.

2.2 Sources of gender bias in MT

Considering the complexity of implications of gender bias in MT described above, it can be assumed that this problem goes beyond the scope of machine translation. MT and NLP models are considered to exemplify unwanted gender biases present in society (Bolukbasi et al., 2016; Hovy and Spruit, 2016; Caliskan et al., 2017; Rudinger et al., 2018; Garg et al., 2018; Gonen and Goldberg, 2019; Dinan et al., 2020). Some researchers have also emphasized multidimensionality of gender bias sources, among which, for example, there are such broad categories as pre-existing, technical and emergent bias (Friedman and Nissenbaum, 1996).

Pre-existing bias refers precisely to any asymmetries which are rooted in society at large or which reflect personal biases of individuals responsible for the system development. In the context of NLP, this could also include subtle connotational characteristics that permeate language structure and use, as well as gender imbalances. These are manifested most notably through the generic masculine, in which referents in discourse are considered to be men by default – unless explicitly stated (Silveira, 1980; Hamilton, 1991). This affects affects not only women, but also non-binary people (Barker and Richards, 2015).

Technical bias emerges during data collection, system design, training and testing procedures. If present in the data used by these processes, asymmetries in the semantics of language use and gender distribution are respectively inherited by the output of the MT (Caliskan et al., 2017). Methods of mitigating bias at this stage include careful data curation (Barocas et al., 2019; Paullada et al., 2020; Koch et al., 2021; Bender et al., 2021), paired with analyses of what is acceptable from the social and pragmatic points of view (Sap et al., 2020; Devinney et al., 2020, Hovy and Yang, 2021), as well as credible annotation practices (Waseem, 2016, Gaido et al., 2020).

Emergent bias typically occurs after design completion and includes cases of mismatch between users and system design, loss of relevance due to shifts in context of use. An example of emergent bias in MT might be the inability of a system to preserve the linguistic style of a social group or to assign correct gender to its potential users (Hovy et al., 2020).

2.3 Challenges and bias mitigation strategies

The majority of mitigating strategies address technical bias: some studies considered, for example, model debiasing with the help of both internal components – like gender tags (Vanmassenhove et al., 2018) and debiased word embeddings (Bolukbasi, 2016; Escudé Font and Costa-jussà, 2019) – and external components integrated with the MT model, such as lattice re-scoring modules (Saunders and Byrne, 2020) and black-box injections (Moryossef et al., 2019). Research is also being carried out within the context of training data (Reddy and Knight 2016; Zhao et al., 2017; Webster et al., 2018) and evaluation methods (Rudinger et al., 2018; Zhao et al., 2018) improvement. However, as some experts have pointed out, these efforts follow a more focused approach within NLP, and lack a human-computer interaction component which is crucial for the development of gender-inclusive systems (Savoldi et al., 2021; Monti, 2020).

What is more, within these proposed strategies, with a few notable exceptions (Cao and Daumé III, 2020; Saunders et al., 2020; Sun et al., 2021), the discussion around gender bias has been reduced to the binary dichotomy. Current language models can perpetrate harms such as the cyclical erasure of non-binary gender identities (Uppunda et al., 2021) rooted in model and dataset biases "due to tainted examples, limited features, and sample size disparities" (Dev et al., 2021), which, in turn, result from the exclusion and an underrepresentation of non-binary genders in society (Rajunov and Duane, 2019). Therefore, an additional challenge in addressing gender bias in MT concerns the need in reshaping the understanding of gender in language technologies in a more inclusive manner – a problem which is well documented in the field (Dev et al., 2021; Savoldi et al., 2021; Misiek, 2020).

3. Gender-neutral language

Being centered around such a complex social phenomenon as gender, gender-neutral language has not yet achieved universal understanding. Moreover, there is no consensus concerning the definition of gender-fairness in language, also referred to as gender-inclusive, gender-fair or genderless, while the exact approach really depends on the conceptual model of a language and social group it is aimed at. In this section, we provide an overview of gender-neutral language and strategies in this field.

3.1 Definition and general information

Gender-fair language (GFL) was introduced as a response to linguistic gender asymmetry and as part of a broader attempt to reduce stereotyping and discrimination in language (Fairclough, 2003; Maass et al., 2013). By avoiding unfounded, unfair and discriminatory reference to certain social groups, it helps to reduce unfavorable cognitive and behavioral biases and promotes gender equality (Stahlberg et al., 2007). Past research has revealed that gender-fair forms evoke fewer male representations than masculine generics (e.g. Irmen, 2007) and influence individuals' attitudes and perceptions: for example, they lead to more favorable hiring decisions for women and positively influence women's motivation and self-assessment in job interviews (Horvath and Sczesny, 2016; Stout and Dasgupta, 2011). Ultimately, an overall purpose of gender-fair language is to include everybody, regardless of gender and/or sexuality (Douglas and Sutton, 2014; Sczesny et al., 2016). Given that language not only reflects stereotypical beliefs but also affects recipients' cognition and behavior (Menegatti, 2017), the use of expressions consistent with social groups' gender and self-perception can help prevent reinforcement of a biased belief system and prevent discrimination.

However, while a lot of effort has been put into representing female populations in language, non-binary language use has not received enough attention in academia. New developments aimed at ensuring gender equality in languages are often perceived as *excess*ive, and this especially concerns the cases when people "do not conform to cis-normative standards of femininity or masculinity" (Airton, 2018). Additionally, there is a lack of non-binary studies within the machine translation field, as has been pointed out by a number of researchers (Dev et al., 2021, Savoldi et al., 2021, Misiek, 2020). All these factors might result in the adverse effects described in the previous section, especially given the fact that language has been central to the emergence of non-binary gender identities, as challenging cis-normativity – the idea that linguistic categories such as man and woman are "normal" or "natural" – is at the heart of non-binary thinking (Cordoba, 2020).

Moreover, a number of GFL guidelines developed by major international organizations (such as the UN and the European Parliament) still make no mention of strategies to address non-binary people in language, and focus on discrimination and exclusion of women (Trainer, 2021); existing strategies in ensuring gender-fair language are not always aimed at other social groups apart from males and females (Lindqvist et al., 2019) or are not sufficiently disseminated (Harris et al., 2017; McGlashan and Fitzpatrick, 2018; Zimmer and Carson, 2012).

3.2 Gender-neutral language frameworks

When defining a gender-neutral language strategy, a broader as well as narrower approach can be taken. Firstly, linguistic structures used to refer to the extra-linguistic reality of gender vary across languages (Savoldi, 2021), and their type in terms of grammatical gender system defines the means by which gender-fairness is achieved.

In general, different strategies can be used to make language gender-fair and avoid the detrimental effects of masculine generics. The choice of an appropriate strategy depends on the type of language concerned: there are genderless languages (Finnish, Turkish), where gender-specific repertoire is at its minimum; notional gender languages (Danish, English), which display characteristics of lexical gender (*mom/dad*), as well as a system of pronominal gender (*she/he*, *her/him*); and grammatical gender languages (e.g., German, French, Arabic), where each noun pertains to a class such as masculine, feminine, and, if present, neuter. Grammatical gender languages are also characterized by the semantic assignment of gender markings to human referents and a system of morphosyntactic agreement (Stahlberg, 2007; Savoldi et al., 2021).

A gender-fair strategy that has been especially recommended for notional gender languages (Hellinger and Bußmann, 2003) and genderless languages is neutralization. In the framework of neutralization gender-marked terms are replaced by gender-indefinite nouns (English *policeman* by *police officer*). In grammatical gender languages, gender-differentiated forms are replaced, for instance, by epicenes (e.g., *Staatsoberhaupt,* or *Fachkraft* in German). In contrast, feminization which is based on the replacement of masculine generics by feminine-masculine word pairs (e.g., *Elektrikerinnen und Elektriker)* has been recommended for grammatical gender languages.

Even though feminization increases women's visibility, and hence creates more diverse mental images to whom individuals referred (Stahlberg et al., 2001), previous research is inconclusive regarding whether paired forms can eliminate the male bias (Lindqvist et al., 2019). What is more, while neutralization helps avoid male bias and therefore indirectly takes into account all genders, feminization does not solve the problem with the exclusion of non-binary people. Therefore, recent research has been proposing such approaches as gender-neutrality (which is closer to the idea of neutralization) and gender-inclusivity (del Rio-Gonzalez, 2021). These approaches can be considered as the same concept (Papadimoulis, 2018; Lindqvist et al., 2019; Bonnin, 2021), as different aspects or degrees of the single phenomenon (Sczesny et al., 2016), (EIGE, 2019) or two separate strategies, where the term gender-neutral language (GNL) is used to describe a language which avoids any classification of sex or gender, whereas gender-inclusive language (GIL) explicitly challenges binary notions of gender and recognizes the plurality of identities beyond feminine–masculine dimensions (del Río González, 2021).

Some researchers also distinguish between direct and indirect non-binary language (López, 2019a, 2019b). Indirect non-binary language, or INL, aims to refer to all genders without using gender markers – by employing certain linguistic strategies such as using participles instead of adjectives (*Studierende* instead of *Studenten und Studentinnen*) or the use of epicenes *(el pueblo argentino* or *las personas argentinas* instead of *los argentinos*), which makes it similar to the gender-neutral strategies described above. Direct non-binary language, or DNL, is much more obvious because it uses neomorphemes and neopronouns such as *ze* and *zir*, and this strategy can therefore be considered within the framework of gender-inclusive approach. Both categories are considered to be equally important and deserve the attention of practitioners because, although their main objective is to break the generic conception of the masculine, the two categories convey radically different messages: DNL communicates unequivocally that the author respects and supports non-binary people, while the use of INL is perfect for mixed-gender contexts (López, 2020).

Although the use of new grammatical gender systems and direct non-binary language in general (López, 2020) seems to be a rather controversial decision in translation, one should not lose sight of the fact that language is a marker of social belonging (Cordoba, 2020), and the refusal to recognize any social groups in language can contribute to discrimination and social exclusion (Sczesny, 2016). Increasing the linguistic visibility of non-binary people and women takes on special significance in the case of grammatical gender languages, as countries with this language type were found to reach lower levels of social gender equality than countries with notional gender languages or genderless languages. This suggests that there is a close link between the level of gender asymmetries present in language and societal gender inequalities (Hausmann et al., 2009, Wasserman and Weseley, 2009). Additionally, despite the difficulties in implementation and promotion of gender-fair language, there are general positive trends in the language communities in supporting strategies aimed at linguistic inclusion of different social groups (Hekanaho, 2020). Hostile and negative reactions towards new language trends challenging the binary gender system seem to normalize rather quickly (Sendén et al., 2015), especially with active efforts to raise awareness about the advantages, benefits and importance of gender-fair languages (Sczesny and Koeser, 2014).

3.3 Gender-neutral language in machine translation

The problem of GNL is receiving increasing attention from academia. Studies related to gender bias concern not only trends which could potentially harm women, but also non-binary people – for example, Dev et al., analyze the complexity of gender and its linguistic representation, and provide the results of a survey on gender-related harms associated with language technologies conducted among non-binary persons. Among three common NLP tasks (Named Entity Recognition, Coreference Resolution, and Machine Translation) included in the survey, misgendering was one of the most frequently mentioned issues, and in terms severity of harms machine translation was the cause of major concern (Dev et al., 2021).

Some efforts in the NLP community were mainly aimed at solving a problem of underrepresentation of non-binary individuals in task-specific data sets: for example, Cao and Daumé III (2020 and 2021) introduce a gender-inclusive dataset GICoref for coreference resolution; in MT, Saunders et al. have presented a method of tagging words with target language gender inflection (Saunders et al., 2020). Apart from approaches that incorporate additional meta-data during training and testing, allowing for a controlled generation of gender alternatives (Bau et al., 2019; Habash et al., 2019; Alhafni et al., 2020), research in this area also concerns generation of gender variants or gender rewriting. For example, Sun et al. (2021) and Vanmassenhove et. al (2021) present a rule-based and neural rewriter for the generation of gender-neutral singular *they* sentences; however, research in this area is monolingual and is limited to English-specific gender-neutral writing, and, more specifically, only the *they* pronoun.

Although the underlying goal of works in this field is to provide more possibilities for the users to make their preferred linguistic choices, thereby empowering people and whole social groups "to interact with technology in a way that is consistent with their social identity" (Sun et al., 2021), there are still challenges at the intersection of gender-fair language and machine translation: firstly, there is insufficient real-world data for all the GNL strategies (and, more specifically, neopronouns); secondly, solutions in this field consider non-binary genders as a static third category which exists next to male and female genders (Dev et al., 2021), when in reality it is of a fluid and diverse nature.

4. Conclusion

This literature review lays the groundwork for further research, the purpose of which will be to assess the efficiency of machine translation in relation to gender-neutral language use. To this end, we categorized the gender-neutral language problem in terms of gender bias in machine translation, presented existing approaches to gender-neutral language and provided an overview of different strategies in machine translation aimed at mitigating representational harms caused by a biased system.

References

Airton, L. (2018). The De/politicization of Pronouns: Implications of the No Big Deal Campaign for Gender-expansive Educational Policy and Practice. *Gender and Education,* 30(6), 790–810.

Alhafni, B., Habash, N. and Bouamor, H. (2020). Gender-aware Reinflection Using Linguistically Enhanced Neural Models. In: *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing,* 139–150.

Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., Dalvi, F. and Glass, J. (2019). Identifying and Controlling Important Neurons in Neural Machine Translation. In: *Proceedings of the Seventh International Conference on Learning Representations (ICLR).*

Barocas, S., Hardt, M. and Narayanan, A. (2017). Fairness in Machine Learning. *Nips Tutorial,* 1, 2.

Bender, E. M., Gebru, T., McMillan-Major, A. and Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623.

Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V. and Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS 2016).* NY: Curran Associates Inc., 4356–4364.

Bonnin, J. E. and Coronel, A. A. (2021). Attitudes Toward Gender-Neutral Spanish: Acceptability and Adoptability. *Frontiers in sociology,* 6, 35.

Bourguignon, D., Yzerbyt, V. Y., Teixeira, C. P. and Herman, G. (2015). When Does it Hurt? Intergroup Permeability Moderates the Link Between Discrimination and Self-esteem. *European Journal of Social Psychology,* 45(1), 3–9.

Caliskan, A., Bryson, J. J. and Narayanan, A. (2017). Semantics Derived Automatically from Language Corpora Contain Human-like Biases. *Science,* 356(6334), 183–186. Available at: http://opus.bath.ac.uk/55288/.

Cao, Y. T. and Daumé III, H. (2021). An Analysis of Gender and Bias Throughout the Machine Learning Lifecyle. *Computational Linguistics*, 47(3), 615–661. Available at: https://doi.org/10.1162/coli_a_00413.

Cordoba, S. (2020). Exploring non-binary genders: language and identity. [PhD diss.]. De Montfort University.

Crawford, K. (2017). The Trouble with Bias – NIPS 2017 Keynote – Kate Crawford #NIPS2017. *YouTube.* [Video]. Available at: https://www.youtube.com/watch?v=fMym_BKWQzk&t=10s.

Dev, S., Monajatipoor, M., Ovalle, A., Subramonian, A., Phillips, J. M. and Chang, K. W. (2021). Harms of Gender Exclusivity and Challenges in Non-binary Representation in Language Technologies. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1968–1994.

Devinney, H. Björklund, J. and Björklund, H. (2020). Semi-Supervised Topic Modeling for Gender Bias Discovery in English and Swedish. In: *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, 79–92. Available at: https://aclanthology.org/2020.gebnlp-1.8/.

Dinan, E., Fan, A., Wu, L., Weston, J., Kiela, D. and Williams, A. (2020). Multidimensional Gender Bias Classification. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 314–331.

Douglas, K. M. and Sutton, R. M. (2014). "A Giant Leap for Mankind" But What About Women? The Role of System-justifying Ideologies in Predicting Attitudes Toward Sexist Language. *Journal of Language and Social Psychology*, 33(6), 667–680.

EIGE – European Institute for Gener Equality. (2019). *Toolkit on Gender-sensitive Communication. A resource for policymakers, legislators, media and anyone else with an interest in making their communication more inclusive*. Publications Office of the European Union. Available at: https://eige.europa.eu/sites/default/files/20193925_mh0119609enn_pdf.pdf.

Fairclough, N. (2001) *Language and Power.* 2nd ed. Harlow: Pearson Education.

Ferrer, X., van Nuenen, T., Such, J. M., Coté, M. and Criado, N. (2021). Bias and Discrimination in AI: A Cross-Disciplinary Perspective. *IEEE Technology and Society Magazine*, 40(2), 72–80.

Font, J. E. and Costa-Jussa, M. R. (2019). Equalizing Gender Biases in Neural Machine Translation with Word Embeddings Techniques. In: *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, 147–154.

Friedman, B., and Nissenbaum, H. (1996). Bias in Computer Systems. *ACM Transactions on Information Systems (TOIS)*, 14(3), 330–347.

Gaido, M., Savoldi, B., Bentivogli, L., Negri, M. and Turchi, M. (2020). Breeding Gender-aware Direct Speech Translation Systems. *arXiv.org e-Print arXiv:2012.04955.* Available at: https://arxiv.org/abs/2012.04955.

Garg, N., Schiebinger, L., Jurafsky, D. and Zou, J., 2018. Word Embeddings Quantify 100 years of Gender and Ethnic Stereotypes. In: *Proceedings of the National Academy of Sciences*, 115(16), E3635-E3644.

Gonen, H. and Goldberg, Y. (2019). Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings but do not Remove Them. *arXiv.org e-Print arXiv:1903.03862.* Available at: https://arxiv.org/abs/1903.03862.

Gustafsson Sendén, M., Bäck, E.A. and Lindqvist, A. (2015). Introducing a Gender-neutral Pronoun in a Natural Gender Language: The Influence of Time on Attitudes and Behavior. *Frontiers in psychology*, 6, 893.

Hamilton, M.C. (1991) Masculine Bias in the Attribution of Personhood: People = male, male = people. *Psychology of Women Quarterly*, 15(3), 393–402.

Harris, C. A., Biencowe, N. and Telem, D. A. (2017) What's in a Pronoun? Why gender-fair Language Matters. *Annals of Surgery,* 266(6), 932.

Hausmann, R., Tyson, L. D., and Zahidi, S. (2009). *The Global Gender Gap Report 2009*. Geneva: World Economic Forum.

Hekanaho, Laura. (2020). Generic and nonbinary pronouns: usage, acceptability and attitudes. [PhD diss.]. Helsingfors University, Helsinki.

Hellinger, M., and Bußmann, H. (2001, 2002, 2003). *Gender Across Languages: The Linguistic Representation of Women and Men, Vol. 1, 2, 3.* John Benjamins Publishing Company.

Horvath, L. K., and Sczesny, S. (2016).Reducing Women's Lack of Fit with Leadership? Effects of the Wording of Job Advertisements. *European Journal of Work and Organizational Psychology,* 25(2), 316–328.

Hovy, D. and Yang, D. (2021). The Importance of Modeling Social Factors of Language: Theory and Practice. In: *Proceedings of the 2021 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, 588–602.

Hovy, D. and Spruit, S. L. (2016). The Social Impact of Natural Language Processing. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2 (Short Papers)*, 591–598.

Hovy, D., Bianchi, F. and Fornaciari, T. (2020). "You Sound Just Like Your Father" Commercial Machine Translation Systems Include Stylistic Biases. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 1686–1690. Available at: https://aclanthology.org/2020.acl-main.154/.

Irmen, L. (2007). What's in a (Role) Name? Formal and Conceptual Aspects of Comprehending Personal Nouns. *Journal of Psycholinguistic Research,* 36(6), 431–456.

Koch, B., Denton, E., Hanna, A. and Foster, J. G. (2021). Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). arXiv.org e-Print arXiv:2112.01716.* Available at: https://arxiv.org/abs/2112.01716.

Koeser, S. and Sczesny, S. (2014). Promoting Gender-fair Language: The Impact of Arguments on Language Use, Attitudes, and Cognitions. *Journal of Language and Social Psychology,* 33(5), 548–560.

Levesque, R. J. (2011). Sex Roles and Gender Roles. In: *Encyclopedia of Adolescence*. Springer International Publishing, 2622–2623.

Lindqvist, A., Renström, E. A. and Gustafsson Sendén, M. (2019). Reducing a Male Bias in Language? Establishing the Efficiency of Three Different Gender-fair Language Strategies. *Sex Roles,* 81(1), 109–117.

López, Á (2020). Cuando el lenguaje excluye: consideraciones sobre el lenguaje no binario indirecto, *Cuarenta naipes*, (3), 295–312.

López, Á (2021). Direct and Indirect Non-binary Language in English to Spanish Translation. In: *27th Annual Lavender Languages and Linguistics Conference*, Online, 21–23.

Maass, A., Suitner, C. and Merkel, E. M. (2013). Does Political Correctness Make (social) Sense? In: Forgas, J. P., Vincze, O. and László J., (eds.). *Social Cognition and Communication*. Psychology Press, 345–360.

María del Río-González, A. (2021) To Latinx or not to Latinx: A Question of Gender Inclusivity Versus Gender Neutrality. *American Journal of Public Health*, 111(6), 1018–1021.

Martindale, M.J. and Carpuat, M. (2018). Fluency Over Adequacy: A Pilot Study in Measuring User Trust in Imperfect MT. In: *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Association for Machine Translation in the Americas, 13–25. Available at: https://aclanthology.org/W18-1803.pdf.

McGlashan, H. and Fitzpatrick, K. (2018). "I Use Any Pronouns, and I'm Questioning Everything Else": Transgender Youth and the Issue of Gender Pronouns. *Sex Education*, 18(3), 239–252.

Menegatti, M. and Rubini, M. (2017). Geneder Bias and Sexism in Language. In: *Oxford Research Encyclopedia of Communication*. Oxford University Press, 451–468.

Misiek, S. (2020) Misgendered in Translation? Genderqueerness Polish Translations of English-language Television Series. *Anglica. An International Journal of English Studies,* 29(2), 165–185.

Monti, J. (2020). Gender Issues in Machine Translation: An Unsolved Problem? In: von Flotow, L. and Hālah, K., (eds.). *The Routledge Handbook of Translation, Feminism and Gender*. Abingdon Oxon: Routledge, 457–468.

Moryossef, A., Aharoni, R. and Goldberg, Y. (2019). Filling Gender & Number Gaps in Neural Machine Translation with Black-box Context Injection. In: *Proceedings of the First Workshop on Gender Bias in Natural Language Processing,* 49–56. *arXiv.org e-Print arXiv:1903.03467.* Available at: https://arxiv.org/abs/1903.03467.

Habash, N., Bouamor, H. and Chung, C. (2019). Automatic Gender Identification and Reinflection in Arabic. In: *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, 155–165.

Olson, P. (2018). The Algorithm that Helped Google Translate Become Sexist. *Forbes.* [Online]. Available at: https://www.forbes.com/sites/parmyolson/2018/02/15/the-algorithm-that-helped-google-translate-become-sexist/?sh=d675b9c7daa2.

Papadimoulis, D. (2018). *Gender-neutral Language in the European Parliament.* Brussels: European Parliament.

Paullada, A., Raji, I. D., Bender, E. M., Denton, E. and Hanna, A. (2021). Data and its (Dis)contents: A Survey of Dataset Development and Use in Machine Learning Research. *Patterns*, 2(11), 100336.

Prates, M. O. R., Avelar, P. H. and Lamb, L. C. (2020). Assessing Gender Bias in Machine Translation: A Case Study with Google Translate. *Neural Comput & Applic*, 32, 6363– 6381. Available at: https://doi.org/10.1007/s00521-019-04144-6.

Reddy, S. and Knight, K. (2016). Obfuscating Gender in Social Media Writing. In: *Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Social Science*. Association for Computational Linguistics, 17–26.

Régner, I., Thinus-Blanc, C., Netter, A., Schmader, T. and Huguet, P. (2019). Committees with Implicit Biases Promote Fewer Women When they do not Believe Gender Bias Exists. *Nature Human Behavior,* 3(11), 1171–1179.

Richards, C. and Barker, M. J. (2015). *The Palgrave Handbook of the Psychology of Sexuality and Gender.* Palgrave Macmillan.

Rudinger, R., Naradowsky, J., Leonard, B., Van Durme, B. (2018). Gender Bias in Coreference Resolution. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2 (Short Papers)*. Association for Computational Linguistics, 8–14.

Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N.A. and Choi, Y. (2019). Social Bias Frames: Reasoning about Social and Power Implications of Language. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 5477–5490. *arXiv.org e-Print arXiv:1911.03891.* Available at: https://arxiv.org/abs/1911.03891.

Saunders, D., Sallis, R., and Byrne, B. (2020). Neural Machine Translation doesn't Translate Gender Coreference Right Unless You Make It. In: *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, 35–43.

Savoldi, B., Gaido, M., Bentivogli, L., Negri, M., Turchi, M. (2021). Gender Bias in Machine Translation. In: *Transactions of the Association for Computational Linguistics*, 9, 845–874.

Schiebinger, L. (2014). Scientific Research Must Take Gender into Account. *Nature,* 507, 9.

Schnoebelen, T. (2017). Goal-oriented Design for Ethical Machine Learning and NLP. In: P*roceedings of the First ACL Workshop on Ethics in Natural Language Processing*, 88–93.

Sczesny, S., Formanowicz, M. and Moser, F. (2016). Can Gender-fair Language Reduce Gender Stereotyping and Discrimination? *Frontiers in psychology,* 7, 25.

Silveira, J. (1980). Generic Masculine Words and Thinking. *Women‘s Studies International Quarterly*, 3(2-3), 165–178.

Stahlberg, D., Braun, F., Irmen, L. and Sczesny, S. (2007). Representation of the Sexes in Language. In: Fiedler, K. (ed.). *Social communication*. Psychology Press, 163–187.

Stahlberg, D., Sczesny, S. and Braun, F. (2001). Name Your Favorite Musician: Effects of Masculine Generics and of Their Alternatives in German. *Journal of Language and Social Psychology*, 20(4), 464–469.

Stout, J. G. and Dasgupta, N. (2011). When He doesn't Mean You: Gender-exclusive Language as Ostracism, *Personality and Social Psychology Bulletin*, 37(6), 757–769.

Sun, T., Webster, K., Shah, A., Wang, W. Y. and Johnson, M. (2021). They, Them, Theirs: Rewriting with Gender-neutral English. *arXiv.org e-Print arXiv:2102.06788.* Available at: https://arxiv.org/abs/2102.06788.

Switzer, J. Y. (1990). The Impact of Generic Word Choices: An Empirical Investigation of Age- and Sex-related Differences. *Sex Roles*, 22(172), 69–81.

Tallon, T. (2019). A Century of "shrill": How Bias in Technology has Hurt Women's Voices. *The New Yorker*. Available at: https://www.newyorker.com/culture/cultural-comment/a-century-of-shrill-how-bias-in-technology-has-hurt-womens-voices.

Trainer, T. (2021). The (non) Binary of Success and Failure: A Corpus-based Evaluation of the European Parliament‘s Commitment to Using Gender-neutral Language in Legislation Published in English and Portuguese. [Master's thesis]. University of Porto.

Turchi, M., Negri, M., Farajian, M. and Federico, M. (2017). Continuous Learning from Human Post-edits for Neural Machine Translation. *The Prague Bulletin of Mathematical Linguistics,* 108, 233–244.

Uppunda, A., Cochran, S.D., Foster, J. G., Arseniev-Koehler, A., Mays, V. M. and Chang, K. (2021). Adapting Coreference Resolution for Processing Violent Death Narratives. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 4553–4559. *arXiv.org e-Print arXiv:2104.14703*. Available at: https://arxiv.org/abs/2104.14703.

Vanmassenhove, E., Emmery, C. and Shterionov, D. (2021). NeuTral Rewriter: A Rule-Based and Neural Approach to Automatic Rewriting into Gender-Neutral Alternatives. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, 8940–8948. *arXiv.org e-Print arXiv:2109.06105*. Available at: https://arxiv.org/abs/2109.06105.

Vanmassenhove, E., Hardmeier, C. and Way, A. (2018). Getting Gender Right in Neural Machine Translation. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3003–3008. *arXiv.org preprint arXiv:1909.05088.* Available at: https://arxiv.org/abs/1909.05088.

Waseem, Z. (2016). Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In: P*roceedings of the First Workshop on NLP and Computational Social Science*. Association for Computational Linguistics, 138–142.

Wasserman, B. D., and Weseley, A. J. (2009). ¿Qué? Quoi? Do Languages with Grammatical Gender Promote Sexist Attitudes? *Sex Roles,* 61, 634–643.

Webster, K., Recasens, M., Axelrod, V. and Baldridge, J. (2018). Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. *Transactions of the Association for Computational Linguistics*, 6, 605–617.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V. and Chang, K. W. (2017). Men Also Like Shopping: Reducing Gender Bias Amplification Using Corpus-level Constraints. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2979–2989. *arXiv.org e-Print arXiv:1707.09457.* Available at: https://arxiv.org/abs/1707.09457.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V. and Chang, K. W. (2018). Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2 (Short Papers)*. Association for Computational Linguistics, 15–20. *arXiv.org e-Print arXiv:1804.06876*. Available at: https://arxiv.org/abs/1804.06876.

Zimmer, B. and Carson, C. E. (2012). Among the New Words. *American speech*, 87(4), 491–510.

https://doi.org/10.33919/JCAL.23.1.6

How Much Linguistics in Corpus Linguistics? Review of *Doing Linguistics with a Corpus* by Egbert, J., Larsson, T. and Biber, D. (2020). Cambridge University Press.

Maria Stambolieva

New Bulgarian University

The publication contains 80 pages (with References), organized in eight chapters, and an Appendix. In their abstract to the text, the authors define their work as an attempt to marry traditional corpus linguistics, with its carefully designed and minutely analysed texts, to the modern state of the art – marked by digitalization, abundance of texts and text collections, and wide array of tools. The stated goal is "to explore ways (…) to improve how we approach linguistic research questions with quantitative corpus data".

The **introduction** begins with a provocative parallel between quantitative linguistics and car driving, followed by a quick review of the chapters to come. The parallel with driving stands on the observation that, with recent technological advances, it is increasingly easy to drive a car without knowing much about the engine – just as it has become easy, in corpus linguistics, to use readily available corpora and corpus analysis tools to answer research questions or to obtain results. And just as some understanding of how the car works can be useful in cases of malfunction, the authors insist that basic knowledge of linguistics: an understanding of the nature of a corpus, the linguistic characteristics of the data or the ability to interpret quantitative results are necessary for corpus linguistic analysis. Linguistic skills are involved in the formulation of linguistic research questions and in the interpreting of quantitative results as linguistic patterns. In all the following chapters of the book, this point is illustrated with relevant case studies and emphasized with key points and key considerations.

Getting to Know Your Corpus (Chapter 2) takes up the long-standing, Sinclair vs Biber, discussion on corpus makeup. Attention must of course be paid to both corpus composition and corpus size and, all things being equal, a bigger corpus is an advantage. The reader is nevertheless warned that all things are almost never equal, and decisions on the composition of the corpus should not be taken lightly. Corpus linguists are not, as a rule, interested in how language is used in a corpus as such, but in how language is used in a target register, dialect, etc. – hence the importance of representativeness in corpus design. For the decision process, the authors recommend the following: 1/ careful examination of the metadata and documentation; 2/ examination of the actual texts. The requirement for careful examination of the metadata and documentation before using a corpus for specific research is well supported by the results of a Case study: an investigation of the use of nominalisations and linking adverbials in the target domain of published academic writing, as represented in two subcorpora: the academic sub-corpus of the British National Corpus (BNC_AC) and the academic subcorpus of the Corpus of Contemporary American English (COCA_AC).

The third chapter, **Research Designs: Linguistically Meaningful Research Questions, Observational Units, Variables, and Dispersion,** is a presentation of several topics required to understand how quantitative corpus analysis relates to tangible linguistic descriptions. The two underlying major concepts here are research design and research questions. "Research design" is defined as the way in which quantitative linguistic data is collected and organized. Research questions specify what we want to learn about language use by doing corpus analysis; accordingly, these questions dictate the research design. Conversely, once data has been collected according to a particular research design, it should only be used to answer certain types of linguistic research questions. The importance of research design is exemplified with the investigation of research questions involving dispersion, and supported with a case study on English genitives in a variationist, whole-Corpus, and text-linguistic research. The chapter concludes with the following key considerations: 1/ observational units can be defined at the level of the linguistic feature, the text, or the corpus; 2/ results from a variationist research design have a dramatically different interpretation from those from descriptive linguistic research designs; 3/ the text-linguistic research design has many advantages over the whole-corpus research design.

Chapter 4, **Linguistically Interpretable Variables,** addresses the need to ensure that all variables used in a corpus study are linguistically interpretable. A linguistic variable is interpretable when its scale and values represent a real-world language phenomenon that can be understood and explained. To illustrate the points made in this section, the authors present two short case studies: "Measures of collocation" (Case study 1) and "The linguistic interpretation of "keyness" measures" (Case study 2). Case Study 1 explores the use of concordancing for one of the primary goals of the study of collocation – the study of the extended meanings of words beyond their traditional dictionary definitions. A very clear example is presented: the verb *to cause*, traditionally defined as "make something happen". Corpus research demonstrates that this verb frequently co-occurs with words referring to negative events – hence the extended meaning of the verb: "make something *bad* happen". Another example is an exploration, based on immediate context, of the way *man* and *woman* are characterized in the corpus COCA_AC. In summary, the simple frequency approach to collocation is argued to be more appropriate for the purpose of discourse characterization than statistical collocational measures, as the two produce different results and require different linguistic interpretations. Case Study 2 is a presentation, following Egbert and Biber (2019), of keyword analysis and "text dispersion keyness". Text dispersion keyness is argued to have two major advantages: (1) it takes into account the dispersion of a word across the texts of a corpus and (2) it is more directly interpretable in linguistic terms than traditional measures – because a text is a valid unit of language production, while a corpus is not.

Chapter 5, **Software Tools and Linguistic Interpretability**, presents a central thesis of this work, based on a case study analysis of grammatical complexity measurement – complex nominals. The measure of complexity of nominals is problematic because, among other things, it does not distinguish between pre- and post- modification and between single and multiple modification. The authors conclude that in order to ensure reliable conclusions based on existing corpus-analysis tools, considerable post-processing is needed – involving, for instance, the evaluation of accuracy. The analysis of a number of smaller corpora, while more time and work consuming, yields results that are more accurate and linguistically meaningful and interpretable. Researchers are advised to choose or develop such tools and measures that are linguistically sound and well documented.

The question of what constitutes appropriate statistical methods is the focus of Chapter 6, **The Role of Statistical Analysis in Linguistic Descriptions.** Following examination of Null hypothesis significance testing (NHST) as a statistical paradigm, the authors (while not denying the usefulness of statistical methods) here again stress the importance of staying close to the language data. Language "is, and should remain, the primary focus of corpus linguistic investigations". Sophisticated statistical methods often create layers of distance between corpus researchers and the language data they aim to describe, which could affect negatively the linguistic validity of the results. Put differently, any kind of abstracting away from the language data increases the risk of obtaining linguistically uninterpretable results – which, in turn, is more likely to lead to misinterpretations and unsatisfactory conclusions. The chapter ends with the following key considerations: 1/ because sophisticated statistical methods often force researchers to abstract away very far from the language data, it is important to employ minimally sufficient statistical methods and remain as close as possible to the language data; 2/ NHST should always be complemented by consideration of descriptive statics and effect sizes; 3/ in order to interpret numeric results, conscious effort should be made to return to the language data.

Chapter 7, **Interpreting Quantitative Results**, can be seen as a summary and generalization of the issues discussed in the previous chapters. The authors argue that computational linguistics is still linguistics, and that linguistics is done by linguists. Computers can of course process corpus data, but they cannot interpret them as "meaningful patterns of language use". The following sources for qualitative interpretation of data that linguists rely on are highlighted: (1) linguistic context, (2) text-external context (above all, metadata), and (3) linguistic principles and theories. Usage-based linguistics, which "explores how we learn language from our experience of language" (Ellis, 2019), is quoted as a "good example of a healthy relationship between linguistic theory and quantitative corpus linguistics". The key takeaways from this chapter are: 1/ linguistics is done by linguists, not by computers; 2/ in order to be useful, quantitative corpus linguistic analysis should be coupled with sound qualitative interpretation; 3/ in their interpretation of quantitative corpus findings, researchers should be guided by linguistic context, text-external context and linguistic theory.

In the final chapter **(Wrapping Up)** the authors summarise the motives which led them to writing the book: reinstating linguistics at the center stage of (quantitative) corpus linguistic research and pointing to means to achieve this. The output of quantitative analysis is data. Data "are to information what iron ore is to iron: nothing can be done with data until they are processed into information". Information is contained in descriptions, answers to questions that begin with such words as *who*, *where*, *when*, and *how many*. In other words, "[i]nformation is born when data are interpreted" (Stallings, 1989, 2). Statistical analysis can provide us with data, but that data must be interpreted if it is to be useful for linguistic description. Linguistic research begins with the formulation of meaningful linguistic research questions and the purpose of corpus design is to answer these questions.

Concluding remarks

The book reviewed focuses on important issues related to the role of linguists, linguistic theory and linguistic research questions in modern corpus linguistics – issues which have been by-passed or ignored for some time, and particularly in the last decade. Backed by clear argumentation and illustrated with ample data, this Element – as the authors have chosen to define their text – manages to cover substantial ground against prevailing winds and currents. For the linguists in the profession, it is a godsend. For other researchers in the field, it is a must-read.

References

Egbert, J. and Biber, D. (2019). Incorporating Text Dispersion into Keyword Analyses. *Corpora*, 14(1), 77–104.

Ellis, N. (2019). Usage-based Theories of Construction Grammar: Triangulating Corpus Linguistics and Psycholinguistics. In: Egbert, J. and Baker, P., (eds.). *Using Corpus Methods to Triangulate Linguistic Analysis*. New York: Routledge.

Stallings, W. (1989). *Data and Computer Communications.* 4th ed. New York: Macmillan.

ISSN (online): 2815-4967