Effective writing style imitation via combinatorial paraphrasing

Tommi Gröndahl
tommi.grondahl@aalto.fi
Aalto University
Espoo, Finland

N. Asokan
asokan@acm.org
Aalto University
Espoo, Finland

ABSTRACT
Stylometry can be used to profile authors based on their written text. Transforming text to imitate someone else’s writing style while retaining meaning constitutes a defence. A variety of deep learning methods for style imitation have been proposed in recent research literature. Via empirical evaluation of three state-of-the-art models on four datasets, we illustrate that none succeed in semantic retainment, often drastically changing the original meaning or removing important parts of the text. To mitigate this problem we present ParChoice: an alternative approach based on the combinatorial application of multiple paraphrasing techniques. ParChoice first produces a large number of possible candidate paraphrases, from which it then chooses the candidate that maximizes proximity to a target corpus. Through systematic automated and manual evaluation as well as a user study, we demonstrate that ParChoice significantly outperforms prior methods in its ability to retain semantic content. Using state-of-the-art deep learning author profiling tools, we additionally show that ParChoice accomplishes better imitation success than A^4NT, the state-of-the-art style imitation technique with the best semantic retainment.

1 INTRODUCTION
Freedom of speech and privacy are threatened by advances in the field of artificial intelligence (AI), including natural language processing (NLP). Governments, corporations or other institutions can use NLP techniques to deanonymize whistle-blowers and dissidents [50]. Potential methods include the use of stylometry to identify or profile anonymous authors based on writing style [56, 58]. This, in turn, motivates defence measures based on altering writing style while exerting minimal impact on semantic content [3, 9, 43]. Adopting a term from image classification [17], we call this style transfer. The process of style transfer can consist of several transformations, i.e. changes applied to the input text.

While prior attempts have largely focused on mere writing style obfuscation by simple methods like back-and-forth machine translation [3, 9, 12, 36], recent work has adopted more sophisticated procedures for style imitation, targeting a particular style rather than only concealing the original [49, 54, 55]. These approaches are particularly promising because, unlike many automatic paraphrasing schemes [32, 61], they do not require parallel corpora for training. This means that they can be trained on separate corpora representing the source and target style respectively, without the need to match sentences between them.

Reliable semantic retainment is of utmost importance for any style imitation system to be useful in practice. In addition to using textual similarity measures like METEOR [5], prior work has relied on comparative human evaluations typically obtained via crowd-sourcing [49, 55]. However, merely comparing alternatives without measuring semantic retainment is not useful for evaluating real-world applicability, since none of the evaluated alternatives may successfully retain the original meaning, even if participants systematically preferred one alternative to another. By conducting an extensive analysis of imitations by three state-of-the-art imitation techniques [49, 54, 55] using both automatic and manual measures, we demonstrate that none reliably produce semantically appropriate paraphrases of the original sentence. Based on our manual evaluation on 400 sentences (50 from eight imitation directions altogether), almost all acceptable outputs are exact replicas of the original sentence. When these are discarded, the rate of acceptable paraphrases does not exceed 10%. While avoiding unnecessary changes is desirable, viable style imitation must retain semantics across transformations rather than eschew them altogether. Our results show that prior techniques fall short on this count.

We develop an alternative imitation scheme called ParChoice. As shown in Figure 1, ParChoice first generates a variety of possible paraphrase candidates, from which it then selects the candidate that is closest to the writing style of a target corpus. This selection is based on an independently trained machine learning (ML) model. Importantly, the same paraphrase generation component can be combined with different paraphrase selection models to target distinct writing styles.

While prior work on style transfer via grammatical transformations and paraphrase replacement exists [26–30], both the rules and corpora used have been highly limited. Declined interest in such approaches plausibly reflects a more general shift away from rule-based NLP methods [19, 62]. In spite of this, we show that existing paraphrase resources [15, 16], together with certain grammaticality filters and a small set of additional rules, allow generating large numbers of suitable paraphrase candidates when applied combinatorially. We further integrate this paraphrasing approach with a recent method for producing grammatical transformations, which alter specific grammatical features of the sentence while retaining its other properties [22].

We measure ParChoice’s semantic retainment ability with three tests: automatic evaluation via the METEOR score [5], manual evaluation on subsets of the test sets, and a user study eliciting semantic similarity scores from independent human evaluators. ParChoice outperforms the three baseline techniques in all these tests, especially surpassing them by a wide margin in both human evaluations. Only one baseline (A^4NT [55]) is able to retain the original meaning to any significant degree, which it mostly achieves simply by reproducing the original sentence without transformations. We then compare the imitation performance of ParChoice and A^4NT in eight imitation directions using two state-of-the-art deep learning author profilers, yielding 16 test settings altogether. ParChoice achieves better imitation results in 10. These evaluations demonstrate that
ParChoice performs best compared to the state-of-the-art in terms of retaining semantics while achieving imitation success. We summarize our contributions below.

- We present ParChoice: a general style imitation technique that is built to maximize semantic retainment across transformations (Section 4).
- Via a systematic empirical comparison of ParChoice with three state-of-the-art baseline style imitation techniques [49, 54, 55] for gender and author imitation (Section 5), we demonstrate that:
  - ParChoice retains semantic information across stylistic changes far more than any of the baseline techniques (Section 6.1)
  - ParChoice’s imitation performance exceeds that of A^4NT: the best baseline technique in terms of semantic retainment (Section 6.2)

2 BACKGROUND

We briefly review prior work on author profiling via stylometry and proposed methods for its mitigation.

2.1 Author profiling as a privacy threat

Stylometry was initiated as a computational task by Mosteller and Wallace [56, 58]. While most studies have focused on traditional machine learning and feature engineering [1, 21, 25, 48, 58, 63], deep learning methods have recently become popular [4, 11, 18, 57], in line with broader NLP trends [19, 62].

Deanonymization attacks [43] can be used by governments, corporations, or other institutions to compromise the privacy of anonymous authors [3, 9, 10, 39]. That deanonymization attacks are realistice concern is underscored by the high accuracy reported in recent stylometry research, especially when the number of author candidates is relatively small (≤ 20) [58]. Moreover, even modest accuracy with a high author count can significantly increase the probability of identification [43], thereby allowing the adversary to narrow down the pool of likely candidate authors for further, possibly manual, evaluation.

Beyond personal identity, more general properties of the author can also be detected from writing style. These include, among others, gender [52, 53], age [53], or political leaning [49]. We pool the detection of such properties under author profiling, and also treat deanonymization as a special type of profiling. We use the term (author) profiler for ML models used for author profiling.

Mitigating author profiling requires style transfer, i.e. transforming the wring style of the original document without changing its semantic interpretation. Prior work has largely focused on what Mack et al. call iterative language translation (ILT) [36], where the original document is translated across one or more intermediate languages back to the original language [3, 9, 12, 14, 36]. ILT has the benefit of being easily available via machine translation (MT) interfaces provided by e.g. Google. Results on its effectiveness against author identification are mixed, some studies reporting success [3, 14, 36] but not all [9, 12]. Semantic retainment is a major challenge, especially across multiple translations [58].

Importantly, ILT cannot be used to aim a particular target style; it merely changes the input document to a style represented by the MT system used, at most obfuscating the style present in the original document. As such, ILT does not constitute a style imitation technique. In the rest of this paper, we focus on style imitation.

2.2 Prior style imitation techniques

Unlike mere style obfuscation, imitation involves enacting only those changes that are likely to take author profiling to the intended direction [55]. We summarize recent research on the topic. For further details we refer to the papers cited.

Rule-based methods Manually programmed grammatical transformations and paraphrase replacement from pre-existing knowledge bases (e.g. PPDB [16], WordNet [41]) provide alternatives to ILT as simple transformation techniques [26–30, 52]. A considerable advantage over ILT is that such rules can be controlled and hence specified to imitate a particular target style. As an example, Karadzhov et al. [26] apply various transformations to the input document in order to steer its stylistic features toward the average values in the author profiler’s training corpus.

Stylistic MT MT can also be used to translate directly between styles in the same language, if parallel corpora are available for training. Previous work on stylistic MT includes paraphrasing Shakespeare as modern English [61], and automatic text simplification by translating text to a more easily readable variant [60]. While its use in style transformation has been rare, positive results were reported by Day et al. [14]. However, its central shortcoming is the reliance on parallel corpora matching the source and target styles, which are rarely available. More recent style imitation research has centered around methods that require no parallel corpora for training, and we focus on these in the remainder of this paper.

Cross-aligned autoencoder (CAE) The main idea behind state-of-the-art approaches to style imitation [49, 54, 55] has been to generate a style-neutral encoding of the original sentence, which serves as the input to a style-specific decoder. Shen et al. [54] trained an autoencoder using a method called cross-alignment for calibrating the distributions between the encodings derived from different source styles. The encoder produces a latent variable from the input sentence, and the decoder generates the target sentence from this content variable together with the target style feature. Shen et al. implemented cross-alignment by adversarial training over generated samples in both styles.

\(^1\)Our experiments and all studies reviewed here are conducted on English datasets.

Broadening the methods to cover other languages is an important venue for future research.

\(^2\)https://translate.google.com/
Back-translation (BT) As an alternative to CAE’s alignment method, Prabhunoye et al. [49] produce the latent content variable via a pre-trained MT system. They first translate the original English sentence to French, and produce the content variable with the French-English encoder. They then train separate decoders that take this encoding as an input and produce sentences in a particular writing style, as evaluated by an independent (pre-trained) author profiler. BT can thus be considered as a more sophisticated variant of ITL with style-specific decoding allowing imitation of a particular target style. Like ITL, it builds on the notion that MT tends to neutralize stylistic features [51].

\[ \text{A}^4 \text{NT} \] Shetty et al. [55] make use of generative adversarial networks (GANs) in their style imitation system titled Adversarial Author Attribute Anonymity Neural Translation (A^4NT). A GAN consists of two neural networks, where one (the classifier) is trained to classify the outputs of the other (the generator), which in turn is trained to deceive the classifier [20]. Initializing these networks with a pre-trained author profiler and autoencoder, Shetty et al. train the generator to produce sentences classified as the target author as well as preserve the semantic content of the original. The latter is regulated by the probability of reconstructing the original sentence via reverse transformation.

3 ADVERSARY MODEL AND USE SCENARIOS

The entities involved in our use scenario are authors \( \{A_1, ..., A_n\} \) and an adversary. Each author belongs to some class in \( C = \{C_1, ..., C_m\} \). A document \( D \) is written by an author, and is thereby also allocated to a class in \( C \). When we talk about \( D \) "belonging to" a class, we strictly speaking, mean the author of \( D \) belonging to the class. We refer to the true author of \( D \) as \( A_D \), and define a function \( C(A) \) from the author \( A \) to its true class in \( C \). By definition, \( A_D \in C(A_D) \). A profiler \( P \) maps each document \( D \) to a class \( C_i \), which we denote as \( P(D) = C_i \).

When \( A_D \) wants to thwart \( P \) from correctly profiling \( D \) as belonging to its true class \( C(A_D) \), \( A_D \) can apply a series of transformations to \( D \), producing \( D^T \). Since \( D^T \) is produced by \( A_D \), its true class is also \( C(A_D) \). Style transfer is successful if \( P \) maps \( D^T \) to a different class despite this. We further distinguish between obfuscation and imitation, where the former succeeds if \( P \) profiles \( D^T \) incorrectly, and the latter if \( P \) maps \( D \) to a particular target class \( C_{tgt} \). In style imitation, sentences in \( C_{tgt} \) are written in the target style. We provide definitions below.

- \( D_T \) succeeds in obfuscating \( C(A_D) \) for profiler \( P \) if:
  1. \( P(D) = C(A_D) \)
  2. \( P(D^T) \neq C(A_D) \)

- \( D_T \) succeeds in imitation \( C_{tgt} \) for profiler \( P \) if:
  1. \( P(D) = C(A_D) \)
  2. \( P(D^T) = C_{tgt} \)

So far we have considered success in obfuscation or imitation as properties of the transformed document \( D^T \). We can broaden these to cover transfer/imitation techniques themselves, which succeed in these tasks if they produce transformed documents that succeed in them. This makes the success of an imitation technique probabilistic instead of a binary property: it can be estimated based on the extent to which applying the imitation increases the probability of \( P(D^T) \neq C(A_D) \) (obfuscation) or \( P(D^T) = C_{tgt} \) (imitation).

When \( |C| > 2 \), the distinction between obfuscation and imitation is unambiguous, the latter being a variant of the former. If \( |C| = 2 \) (as in our test cases), we consider obfuscation as successful if the probability of correct profiling is reduced to a random level, and imitation as successful if there is a significantly above random probability of misprofiling. Since there is no definitive criterion for what should constitute "significant", the boundary between obfuscation and imitation is somewhat blurred in two-class datasets.

In the absence of style transfer, the adversary has access to \( D \) written by \( A_D \), and the profiler \( P \) that maps each document to some class in \( C \). He applies \( P \) on \( D \), and succeeds if \( P(D) = C(A_D) \). In the imitation scenario, the adversary is only allowed to access \( D^T \). In this paper, we only consider profiling by automatic methods, where \( P \) is a machine learning (ML) model trained on data derived from texts belonging to the classes in \( C \). We therefore assume that the adversary has access to such training data. We call this the (author) profiler dataset.

We further assume \( A_D \) to access example documents from two corpora to train the imitation method: one from the true class \( C(A_D) \) and another from the target class \( C_{tgt} \). These corpora constitute the surrogate dataset. The surrogate dataset is distinct from the author profiler dataset, but we assume them to correspond in labels and distribution. \( A_D \) uses the surrogate dataset to train a surrogate profiler, which is required to conduct the imitation.

4 DESIGN OF PARCHOICE

Figure 2 shows an overview of the ParChoice pipeline. It consists of two stages: (i) paraphrase generation, which takes an input document and generates a set of paraphrase candidates, and (ii) paraphrase selection. To realize the paraphrase selection stage, ParChoice uses a surrogate profiler trained on the surrogate dataset. In this section, we explain each stage in detail, and discuss the parameters that can be used to tune the performance of ParChoice in speed, imitation performance, and semantic retention.

4.1 Paraphrase generation

The paraphrase generation stage contains three modules: grammatical transformations, paraphrase replacement, and adding typos. The modules are always applied in this order, but the paraphrase replacement loop allows multiple iterations, the number of which is specified as a parameter. In Sections 4.1.1–4.1.3 we motivate and describe each module.

4.1.1 Grammatical transformations. This module consists of a parser that maps input sentences into representations of grammatical and semantic structure, and a pre-trained neural machine translation (NMT) model that generates English sentences from these representations. Grammatical transformations form a challenging task for NLP, and not many prior works have tackled them directly. Outside of highly task-specific rule-based approaches [2, 6, 7], we are aware of only two general techniques proposed for creating grammatical transformations [22, 33], one of which we used in ParChoice.

Similar to the A^4NT method for style transformation [55], Logeswaran et al. [33] trained an encoder network to produce a latent content representation of the input sentence, and a decoder
network to generate sentences from the content representation together with a set of target attribute features. They enforced content retention by reconstruction losses from the original sentence’s content representation (autoencoding loss), the decoded sentence (back-translation loss), and their combination (interpolated reconstruction loss). The transformed attributes were represented as Boolean vectors, and included e.g. sentiment (positive/negative), tense (past/present), voice (active/passive), and negation. However, Logeswaran et al. only measured the accuracy of the target class, not semantic retention. Their method is also a more generic text transformation framework, whereas we wanted to focus on grammatical transformations in particular.

Gröndahl and Asokan [22] use a similar technique, with the important distinction that instead of an encoded latent representation they use a symbolic format generated from the input sentence’s dependency parse [59]. Since this intermediate representation is human-readable, it can be modified with targeted transformation rules. The transformed sentence is then generated with an NMT network trained to translate the representation back to English. This technique allows producing grammatical transformations between multiple class pairs (declarative-question, present-past, active-passive etc.) in any direction or combination. Gröndahl and Asokan further demonstrate the method to maintain a high level of content preservation as measured by back-transformation overlap with the original sentence as well as manual evaluation. Given the demonstrated success of this method in generating a large variety of grammatical transformations, we adopted it in ParChoice.3

We restricted grammatical transformations to those having a minimal semantic effect. With additional appended material we were able to utilize transformations between affirmed and negated sentences, as well as between questions and declaratives. We also used voice transformation (active-passive) in both directions whenever it was available. Examples are shown in Table 1.

**Voice** Transitive verbs can appear either in the active or passive voice, which impacts the semantic interpretation of the grammatical subject. The direct object of an active clause is expressed as the subject in the corresponding passive clause, and the subject of an active clause is optionally expressed in the passive via the preposition by. We produced both active and passive versions of all sentences that contained both required arguments.

**Negation** In addition to using the negative particle not/n’t, an affirmative sentence can be negated by embedding it within a main clause that states the falsity of its subordinate clause. We make use of this fact by first producing affirmative versions of negated input sentences, and subsequently wrapping them in the context “I do not/n’t think/believe (that)...”. We chose this prefix due to its fluency in comparison to other options, such as “It is not the case that...”. Of course, different alternatives could be chosen depending on the nature of the input data. The first person narrative is appropriate for our test data, as restaurant reviews, blogs, and political speeches are typically written from a personal standpoint (cf. Section 5.1.1).

**Questions** To paraphrase a polar (yes-no) question, we first transformed it to a declarative variant, which we then appended to the prefix “Is it (true) that...?” For negative questions, we additionally produce the affirmed declarative variant embedded in “Is it (true) that...” and “Isn’t it (true) that...”. We employed voice transformation when generating the declarative sentences.

4.1.2 Paraphrase replacement. After grammatical transformations we applied paraphrase replacement using both simple rules and two external paraphrase corpora: PPDB [16] and WordNet [41]. Simple rules are applied first, after which the loop PPDB→WordNet→simple is iterated for a specified number of times. Each iteration enact one or more transformations on the input

---

3The authors of the original paper [22] made their model and training data available to us. The model consists of an encoder and a decoder network, both of which are two-layer LSTMs with 600 hidden units in each layer. The intermediate representations are sequences of 928-component vectors, and use pre-trained GloVe embeddings [47] for representing words. We refer to the original paper for further details. The training corpus consists of 8.5 million English sentences of the maximum length of 20 words, derived from the following sources: Stanford parallel corpora (English only) [34, 35] (https://nlp.stanford.edu/projects/nnmt/), OpenSubtitles 2018 (http://opus.nlpl.eu/OpenSubtitles2018.php), Tatoeba (English only) (https://tatoeba.org), SNLI [38] (https://nlp.stanford.edu/projects/snli/), SICK [38] (http://clic.cimec.unitn.it/composes/sick.html), Aristo-mini (December 2016 released) (https://www.kaggle.com/allenai/aristo-mini-corpus), and example sentences from WordNet [41] (http://www.nltk.org_/modules/nltk/corpus/reader/wordnet.html). We also used this corpus for obtaining inflection information for WordNet lemmas, as explained in Section 4.2.

4A verb is transitive if it takes both a subject and a direct object in the active voice.
We apply a small set of rules to produce simple alterations with little to no semantic effect. In addition to supplying paraphrases on their own, such rules are useful for broadening the range of inputs for further processing stages, especially paraphrase replacement from PPDB.

Indefinite pronouns with the suffix -body are interchangeable with pronouns with the suffix -one. We allow any combinations of the following synonyms in the original sentence:

\{(anybody, anyone) \{somebody, someone\} \{nobody, no one\}\}

Grammatical elements that modify verbs in terms of possibility or judgement (can, should, must, etc.) constitute the class of modal auxiliaries. Some of these are sufficiently equivalent to allow replacement in most contexts, such as can and could. However, some are only equivalent in either an affirmed or a negated context, but not both. The context is negated if the auxiliary precedes a negation (not/n’t), and affirmed otherwise. For instance, Mary could win is (roughly) equivalent Mary may win, but Mary could not win differs substantially from Mary may not win. Hence, could and may are (roughly) equivalent in an affirmed context but not in a negated context. The following sets represent the equivalent modal auxiliary groups we use:5

**Affirmed context**

\{might, may, could, can\} \{should, ought, must\} \{will, shall\}

**Negated context**

\{can, could\} \{should, ought, must\} \{will, shall\}

English allows the negation, certain inflections of the copula verb (am, are, is), or the auxiliary have to be expressed in a contracted form. Discarding the contraction’s due to its high ambiguity,6 we consider the following as paraphrases:

\{not, n’t\} \{am, ’m\} \{are, ’re\} \{have, ’ve\}

We produce both contracted an non-contracted variants of each, as allowed by their grammatical context.7

Finally, we maintain that removing commas only has a minor impact on semantic interpretation,8 and therefore allow any comma in the sentence to be optionally removed. As a very minor transformation, comma removal is mostly relevant for increasing the range of options for paraphrase replacement from PPDB.

**PPDB** The Paraphrase Database (PPDB) is automatically collected from parallel bilingual corpora based on the assumption that paraphrases are often translated to the same target sentence [16]. It contains more than 220 million paraphrases, which have subsequently been annotated for semantic relation type (equivalence, entailment etc.) and ranked with a supervised regression model [46]. We restricted replacements to the equivalent class, resulting in 245691 phrase pairs altogether.9

---

5We append ought with the preposition to (following the negation in negated contexts), and conversely remove to if ought is replaced with another auxiliary.

6The clitic’s can replace both is and has, in addition to functioning as a homophonous possessive suffix. There are also fairly complex restrictions regulating its appearance with is or has that make it challenging to use in simple paraphrase replacement.

7We only produced the contraction ‘ve after a pronoun in {I, you, we, they}, and the contracted negation n’t after an auxiliary in {is, are, was, were, have, has, had, we (variant of will), must, should, need, ought, could, can, do, does, did}. We only replaced ‘ve with the non-contracted version have and not vice versa, in order to avoid the ungrammatical transformation of a non-auxiliary have to ‘ve.

8We are aware of counterexamples like Let’s eat, I grandma or I love cooking and I dog and playing golf. However, we maintain that the intended meaning is not irreversible in such cases, but merely less evident due to ambiguity. Removing commas may thus introduce alternative readings, but does not remove the original interpretation.

9We derived these from the so-called “PPDB-TLDR” version, which contains lexical and phrasal paraphrases from the XL class, within the ranking scale between S to XXXL
Some paraphrases contain additional information about the syntactic context in which they can be placed. Using the Python NLP library Spacy\textsuperscript{10} \cite{spacy} to derive syntactic information from the original sentence, we only allowed replacement if the context matched that specified in the PPDB entry. Some paraphrases lack context information, but still specify the part-of-speech (POS) of the head word of the phrase.\textsuperscript{11} In such cases, we used the POS tag to restrict paraphrase replacement. Based on manual evaluation on independent validation sets, these grammatical restrictions highly improved the grammatical status of the resulting transformations.

PPDB operates on the level of raw text, where words appear in inflected form. WordNet \cite{wordnet}, in contrast, is a manually built knowledge base of word senses, which represent possible word meanings and contain various properties including synonyms, hyper/hyponyms, definitions, and example sentences. All words occur in an abstract lemma form, and no inflection information is included. While WordNet has often been used in prior rule-based style transfer \cite{chen2018rouge,lin2020text,ren2021style,gu2021paraphrase}, this format is a major obstacle for its direct application.

Selecting one word sense to represent a word in the sentence is known as word sense disambiguation \cite{grondahl2019style}. For this task, we used the simple Lesk algorithm from Python’s WSD library,\textsuperscript{12} which is a variant of the Lesk algorithm \cite{lesk1986} based on overlap in word sense properties (definition, example sentences, hypernyms etc.). To enact inflection, we created a dictionary from lemmas to their surface manifestations with different POS-tags, deriving these from a large text corpus.\textsuperscript{13} To produce a paraphrase, we inflected all synonyms of the word sense (attained via simple Lesk) based on the POS-tag of the original word, and replaced the original word with each of its inflected synonyms.

### 4.1.3 Typos

Typographical errors are an important aspect of stylistic variation \cite{biber2008variation,tabakowska1994typos,grondahl2019style}. However, given the vast number of possible misspellings and their varying effects on readability, introducing them randomly is not justifiable. Instead, we introduced typos that appear in the target corpus of the surrogate dataset. For obtaining typos we used the Python port of SymSpell.\textsuperscript{14} applying it to the target corpus and storing a dictionary from correct spellings to their possible misspellings. We additionally spell-checked the original sentence and included original typos to this typo dictionary.\textsuperscript{15} This allowed either retaining original typos or correcting them, depending on their effects on paraphrase selection.

### 4.2 Paraphrase selection

A selection from the set of paraphrase candidates is done twice in the ParChoice pipeline. These selections are based on the surrogate profiler, which is a ML model trained on the surrogate dataset to classify between the source and target writing style. First, among the candidates produced by the paraphrase replacement module, only those that the surrogate profiler maps to the target class are retained for potential further iterations. Second, after paraphrase generation, ParChoice selects the paraphrase candidate which is assigned the highest target class probability by the surrogate profiler.

**Surrogate profiler** As our surrogate profiler, we used a logistic regression (LR) classifier with word unigrams as input features. Notably, this architecture is much simpler than the deep learning author profilers we used for evaluating imitation performance (cf. Section 6.2). If deep learning profilers can be deceived using only an LR surrogate profiler, this indicates that imitation cannot be thwarted simply by increasing author profiler complexity.

**Hyperparameters** There are three hyperparameters that we can use to control ParChoice’s paraphrase generation stage. Each has different expected effects on imitation success and semantic retainment. Additionally, increasing the number of candidate paraphrases naturally slows down the speed of ParChoice.

The first parameter concerns the truncation of the candidate set after every iteration of the paraphrase replacement loop. We call this maximum #candidates. As the truncated subset is chosen randomly, increasing it has no detrimental effect on semantic retainment. Since it also improves the likelihood of finding an optimal imitation, its only downside is the slowing down of processing speed.

The second parameter is the number of times the paraphrase replacement loop is iterated. We call this #iterations. Unlike maximum #candidates, #iterations has a negative impact on semantic retainment in addition to processing speed. This is because every iteration builds on top of prior paraphrase candidates, and hence exerts more drastic modifications to the original sentence. However, additional iterations can improve imitation success by generating candidates which would otherwise not be produced.

Finally, we also filter paraphrase candidates at the end of each loop iteration based on edit distance measured between the candidates and the original document. Since edit distance is likely to correlate negatively with semantic retainment, an upper bound on edit distance reduces the probability of semantically unviable candidates. However, it also decreases more drastic transformations that might be needed for successful style imitation.

The table reports the expected effects of increasing each of the three hyperparameters, where “+” marks an increase, “−” a decrease, and “+/−” a lack of systematic correlation to either direction.

| Parameter     | Effects of increasing |
|---------------|-----------------------|
|               | Semantic retainment   | Imitation | Speed |
| Maximum #candidates | +/−  | +  | +  |
| #iterations    | −  | +  | +  |
| Maximum edit distance | −  | +  | +  |

Table 2: Effects of hyperparameter choice
5 EXPERIMENTS

In this section we describe the experiments of empirically comparing ParChoice with three state-of-the-art style imitation techniques as the baseline. We applied each technique to four two-class datasets in both directions, and tested imitation success against two deep learning author profilers. We review the test settings in Section 5.1 and the evaluation metrics in Section 5.2.

5.1 Test settings

This section introduces the datasets, baseline models, and ParChoice hyperparameters we used in our tests.

5.1.1 Datasets. We used four two-class corpora, two of which are labeled by gender and two by author identity. The gender corpora and one author corpus are used in two of the papers presenting the baseline methods [49, 55]. The remaining author corpus is derived from one of the gender corpora [53], and has not been used in prior style imitation research. Here we describe each dataset and contrast some of their main characteristics. Size- and length-related information is presented in Table 3. All datasets can be downloaded from the baseline models’ project pages on GitHub.\(^\text{16}\)

**Yelp gender (YG)** Originally collected by Reddy and Knight [52], this corpus consists of restaurant reviews labeled by the gender attribute (based on the author’s proper name). It was used by Prabhumoye et al. in their BT method described in Section 2.2 [49]. The datapoints are typically complete and grammatically correct sentences containing mostly common English terminology and only rarely typos.

**Blog gender (BG)** The blog dataset, collected by Schler et al. [53], contains blog posts labeled by authorship, gender, age, and zodiac sign. In contrast to YG, colloquial expressions, abbreviations (e.g. LOL, OMG), and interjections (e.g. *ouch, sigh*) appear often in this dataset. This makes it more challenging for paraphrasing, as both dependency parsing (required for grammatical transformations) and paraphrase databases (PPDB, WordNet) are built for mostly grammatically correct and typo-free input. Another important contrast to YG is the much larger vocabulary size, which is indicative of the wider range of terminology used in blogs in comparison to restaurant reviews.

**AB** Blog posts

**TO** Political speeches

Table 3: Dataset properties

| Name | Data type       | Class labels | Sentences | Vocabulary size | Sentence length | Mean |
|------|-----------------|--------------|-----------|-----------------|-----------------|------|
| YG   | Restaurant reviews | male, female | 3208136   | 19996           | 4               | 101  | 18.2 |
| BG   | Blog posts      | male, female | 3379779   | 264966          | 2               | 30   | 12.8 |
| AB   | Blog posts      | author 1 ("Alice") author 2 ("Bob") | 31319     | 16498           | 1               | 110  | 11.2 |
| TO   | Political speeches | Trump Obama | 39925     | 13231           | 2               | 294  | 17.81 |

Alice-Bob (AB) From the blog dataset, we extracted the two authors with the most sentences, and used them to build a stand-alone dataset for author profiling. However, while BG only contains sentences with 30 words or less, we used all sentences by these authors to increase dataset size. Therefore the maximum sentence length is higher in AB than in BG, as shown in Table 3. One author is a female in the age range 13 - 19, and the other male a in the age range 23 - 40. For convenience, we call them Alice and Bob, respectively. Of our four datasets, only AB has not been used in prior style imitation research.

**Trump-Obama (TO)** This dataset includes speeches by the two most recent US presidents: Barack Obama and Donald Trump. Collected by Shetty et al. [55], it originally contains 372 speeches and 65485 sentences overall. However, it is highly imbalanced, and we were able to significantly improve profiling accuracy by truncating the larger class (Obama) to the same size as the smaller. We used this balanced version in our experiments. As this dataset consists of prepared speeches, it is generally high in grammatical quality and contains few colloquialisms, abbreviations, or interjections.

5.1.2 Baseline style imitation techniques. We review the technical details involved in each baseline we compared against ParChoice. Some were obtained as pre-trained models from the projects’ GitHub pages, and the rest we trained ourselves based on the descriptions in the original papers. The basic functions of the three baselines were outlined in Section 2.2.

**Cross-aligned autoencoder (CAE)** The authors of the CAE paper [54] do not use any of the datasets we experimented on. However, in Prabhumoye et al.’s paper presenting BT [49], CAE is used as a baseline against BT. The authors trained both on the YG dataset, and compared their imitation performance as well as human preference on sentiment retention. In a user study, Prabhumoye et al. asked human annotators the question “Which transferred sentence maintains the same sentiment of the source sentence in the same semantic context?” [49]. Additionally, the authors manually evaluated the fluency of 60 randomly chosen sentences. While CAE performed slightly better on gender imitation than BT (60% vs. 57% success), BT outperformed CAE in both sentiment retention and fluency. In the user study, 43% of the BT-generated sentences were deemed superior in comparison to only 15% of the CAE-generated sentences (41% belonging to the “no preference” group). In the fluency measurement, BT received an average score of 2.81 out of 4, in comparison to CAE’s 2.42.

\(^{16}\)Yelp and TO: https://github.com/rakshithShetty/A4NT-author-masking/blob/master/README.md

AB is derived from BG.
Back-translation (BT) Prabhumoye et al. [49] provide the BT model trained on YG on the project’s GitHub page.\footnote{https://github.com/shrimai/Style-Transfer-Through-Back-Translation} We used this for our YG experiments, and trained BT ourselves for other datasets. When training on AB, the resulting model exhibited very poor semantic retainment due to the small size of the dataset, only repeating a few words. To add lexical variation, we initialized the model’s weights with those of the original French-English translator not tailored for any target style. This markedly improved the model’s behavior by making lexical choices more diverse and reducing repetition. We did not initialize weights for other datasets due to their larger size.

A^4NT The GitHub page of A^4NT\footnote{https://github.com/rakhithShetty/A4NT-author-masking/blob/master/README.md} includes pre-trained models for BG and TO.\footnote{The pre-trained A^4NT TO-model has been trained on the original Trump-Obama Dataset, whereas we used a balanced version for training our profilers and other imitation techniques (CAE, BT), as explained in Section 5.1.1. For comparison we also trained A^4NT on our balanced version of TO, but used the pre-trained model in our tests as it performed better in imitation.} We trained A^4NT ourselves for YG and AB. While we used the full dataset to train the initial classifier and generator for YG, hardware limitations required us to truncate YG for training the GAN. We used a subset of 100000 sentences in this stage.

In the paper presenting A^4NT, Shetty et al. experimented on BG, TO, and blog author age (the same corpus as BG but with different labels). On all corpora, they managed to reduce classification accuracy (combined from both classes) from above 0.50 below 0.50. They further measured semantic retainment via the METEOR score\footnote{[5]}, and conducted a user study where they compared A^4NT to ILT via Google Translate (cf. Section 2). A^4NT received a high evaluation score of 4.51 on a 0 – 5 scale, which superseded ILT’s score of 4.16.

5.1.3 ParChoice. We applied the ParChoice pipeline as described in Section 4 to test sets derived from YG, BG, AB, and TO. As hyperparameters (cf. Section 4.2) we used one paraphrase replacement iteration, the maximum candidate set size of 1000, and the maximum edit distance of 10 (cf. Section 7 for discussion).

As mentioned in Section 4.2, our surrogate profiler was a word-based LR classifier. We used 10000 as the maximum number of features, and trained the surrogate profiler on a separate surrogate dataset for each test setting. Importantly, this surrogate dataset was always independent of the data used to train the author profiler. In this respect, ParChoice is less restrictive than any of the three baseline methods, where the same data is used to train the style imitation system and the author profiler.

We constructed the surrogate datasets from the original test sets, not used for training the author profiler. For consistency across experiments, we always used 15% of the author profiler training set size as the surrogate dataset size. We then conducted our tests on a separate test set unseen by both the author profiler and the surrogate profiler. Training and test set sizes are shown in Table 4.

### Table 4: Training and test set sizes

| Dataset | Training set sizes | Test set size |
|---------|--------------------|---------------|
| YG      | 2577862            | 5000 × 2      |
| BG      | 2353325            | 5000 × 2      |
| AB      | 25319              | 588 × 2       |
| TO      | 20860              | 2334 × 2      |

5.2 Evaluation

We measured the effectiveness of each style imitation technique on two fronts: semantic retainment and imitation success.

5.2.1 Semantic retainment. We evaluated semantic retainment using multiple automatic and manual metrics. The former were measured for the entire test sets and the latter for smaller subsets. For manual evaluations, we used both our own analysis and a user study by independent human evaluators to estimate the imitation techniques’ ability to retain the original meaning.

Automatic evaluation The most basic estimate of semantic retainment is the n-gram overlap with original sentence, standardly measured via the BLEU score developed for evaluating MT\cite{13, 45}. However, BLEU penalizes any divergence from the original text, which is undesirable for our purposes. As a more sophisticated metric, the METEOR score\cite{5} uses additional paraphrase tables, which makes it more appropriate for our task. We further provided two METEOR scores: one for all imitations, and another for those that are non-identical with the original sentence. The latter is more relevant for evaluating semantic retainment when transformations occur, which is our main interest.

Manual evaluation For the eight imitation directions, we manually compared a 50-sentence subset of the original test sentences to their imitated counterparts, and classified each imitation to one of the following classes:

- identical to the original sentence
- paraphrase of the original sentence
- omission: contains only part of the original sentence
- other error(s): inappropriate changes or additions

User study In addition to our own manual evaluation, we conducted a user study on 20 people who were not involved in the development of ParChoice or any of the baselines. 25% of the users were native English speakers. 55% were female and 45% male. 60% were 20–30 years old, and 40% were 30 or older. 90% had a university degree, most often on the Master’s level (65%).

We administered the study as an online questionnaire. Each user was allocated a unique set of 20 sentences drawn from the YG dataset. They were shown the original sentence along with its imitation to the opposite gender’s writing style by all imitation techniques we used (in a randomized order): CAE, BT, A^4NT, and ParChoice. We only used imitations that were non-identical to the original sentence. Replicating the method used by Shetty et al.\cite{55} for evaluating A^4NT, we asked the users to compare all four imitation variants to the original, and rate them on a 0 – 5 scale based on how close the meaning of the variant was to that of the original. We gave the users no additional information about the test sentences being computer-generated for the purpose of style imitation.
### 5.2.2 Imitation success

To evaluate imitation success, we trained two author profilers to classify the original and imitation test sets produced with the baselines and ParChoice. We use the following state-of-the-art NLP techniques to construct our profilers: a long short-term memory network (LSTM) derived from Shetty et al. [55], and a convolutional neural network (CNN) derived from Prabhumoye et al. [49]. Both use words as input features. The code we used for training these is available on the respective papers’ GitHub pages.\(^{20}\)

### 6 RESULTS

This section presents the results on semantic retainment and imitation success. Overall, ParChoice exhibits a much higher semantic retainment than the baselines based on both automatic and manual measurements, and a superior imitation performance to A\(^{4}\)NT, which is the only viable baseline in terms of semantic retainment.

#### 6.1 Semantic retainment

We review the results from the three tests we used to measure semantic retainment: automatic evaluation, manual evaluation, and the user study. METEOR scores are presented in Table 5, manual evaluation results in Table 6, and user study results in Table 7.

**Automatic evaluation** ParChoice and A\(^{4}\)NT always clearly outperform CAE and BT in METEOR. Especially on the smaller datasets (AB, TO), CAE and BT produce very poor scores (< 10), which indicates that they bear little to no resemblance to the original sentence. The score of ParChoice exceeds that of A\(^{4}\)NT in most cases, and in all cases when identical sentences are discarded. Hence, ParChoice more often produces paraphrases recognized by the METEOR paraphrase table. This indicates that ParChoice is able to retain the original meaning better than any of the baselines. As will be seen, results from our manual evaluations and the user study strongly corroborate this hypothesis.

**Manual evaluation** Table 6 presents results from our manual evaluation on 50 sentences for all four datasets to both directions. The last two columns contain the most important part: the semantic retainment rate, which shows how often applying the imitation technique resulted in an acceptable paraphrase of the original, based on our judgement. As with METEOR, we distinguish between the semantic retainment rate of all evaluated sentences and the rate of those sentences that are non-identical to the original. The latter indicates how often the imitation technique is able to produce acceptable paraphrases of the original sentence without simply reproducing it. As can be seen from the table, only ParChoice produced these systematically, the baselines achieving only a 0% – 10% retainment rate as opposed to ParChoice’s 37% – 64%.

Of the baseline techniques, CAE and BT produced practically no paraphrases, and rarely reproduced the original. This is especially true in the small datasets (AB, TO), where imitations bore no resemblance to the original sentence and simply repeated certain words prevalent in the target corpus. In A\(^{4}\)NT the reproduction rate was by far the highest, but non-identical paraphrases remained very rare, as seen in the drastic decrease in semantic retainment rate when only non-identical sentences are considered (last column). ParChoice stands out in clear contrast to all baselines in producing many paraphrases, resulting in the overall semantic retainment rate being by far the highest across all datasets and imitation directions.

Most errors were due to inappropriate word replacements. CAE and BT usually produced novel sentences that only bore a distant lexical and grammatical similarity to the original. In the BG dataset, A\(^{4}\)NT also generated a significant number of omissions, where some words were simply left out from the original and nothing

#### Table 5: METEOR scores (highest results emphasized)

| Data | Direction | CAE | BT | A\(^{4}\)NT | ParChoice |
|------|-----------|-----|----|-------------|------------|
| YG   | female→male | 19.63 | 21.97 | 43.77 | 44.80 |
|      | male→female | 19.04 | 21.54 | 36.47 | 43.81 |
| BG   | female→male | 30.83 | 22.58 | 51.27 | 47.05 |
|      | male→female | 28.11 | 28.17 | 41.94 | 46.14 |
| AB   | Alice→Bob   | 7.72  | 5.56  | 31.21 | 44.60 |
|      | Bob→Alice   | 7.36  | 5.39  | 26.60 | 43.33 |
| TO   | Trump→Obama | 5.94  | 4.89  | 22.55 | 47.95 |
|      | Obama→Trump | 5.19  | 6.19  | 20.78 | 47.14 |

\(^{20}\)LSTM: https://github.com/rakshithShetty/A4NT-author-masking/blob/master/README.md

CNN: https://github.com/shrimai/Style-Transfer-Through-Back-Translation
else was changed. This likely partly accounts for A4NT's high METEOR score in BG. ParChoice's errors were most often caused by inappropriate WordNet synonyms, reducing which would require improving the word sense disambiguation algorithm used to map words to WordNet senses. This is an important consideration in the future development of ParChoice.

Even though all imitation techniques occasionally resulted in grammatical errors, all techniques trained on large datasets (YG, BG) typically produced full English sentences. When trained on small datasets (AB, TO), CAE and BT repeated a small number of words in largely ungrammatical combinations, whereas A4NT produced mostly fluent sentences despite failing in semantic retainment. Uniquely, ParChoice’s did not deteriorate in semantic retainment depending on the training corpus size. This is due to the distinction between paraphrase generation and paraphrase selection, only the latter being affected by the target corpus.

User study Results from the user study are presented in Table 7. All models received grades across the whole spectrum. However,
the highest mean grade is only 2.7 (achieved by ParChoice), which indicates that all imitation techniques had trouble with semantic retention. Still, ParChoice performed significantly better than any of the baselines. Especially CAE and BT had very poor performance, not exceeding 1 in the mean grade (on the scale 0 – 5).

ParChoice particularly stands out in the percentage of imitations that obtained high grades. CAE and BT received the grade 4 or 5 in less than 10% of imitations and A^4NT in 20%, whereas ParChoice achieved 4 or 5 in more than 40% of imitations. Around one fourth of all ParChoice-imitations received the best grade 5, which surpassed the baselines even more drastically (2% in CAE, 3% in BT, and 9% in A^4NT). Finally, the majority of ParChoice-imitations were on the upper half of the six-point scale (3 – 5), whereas the majority of all baseline imitations had the grade 0 or 1.

### 6.2 Imitation

To evaluate imitation success, we measured author profiler accuracy on both classes before and after imitation. However, CAE and BT completely failed in the semantic retention task. In Table 8 we therefore only present a comparison between ParChoice and A^4NT, but provide the full comparison in Appendix A. We consider 0.50 as the imitation limit, and hence an accuracy change from above 0.50 to below 0.50 as successful imitation.

**YG** Both the LSTM and CNN profiler achieved high accuracies in both directions with YG. ParChoice’s performance always exceeded those of A^4NT in YG, and ParChoice succeeded in imitation in all cases except male→female measured with the CNN profiler. In contrast, A^4NT did not cross the imitation limit, at best taking accuracy down to 0.57. Both A^4NT and ParChoice performed better in the female→male direction than conversely.

**BG** Original profiling accuracy was highly imbalanced in BG, only the female class achieving a decent score with either author profiler. Male classification accuracy remained below 0.50 with both profilers. Hence, while both A^4NT and ParChoice decreased accuracy in the male→female condition, this does not satisfy our criteria for successful imitation. Furthermore, A^4NT did not take profiling accuracy down in the female→male direction at all, instead increasing it by a few percent for both author profilers. ParChoice failed to imitate as well, but still decreased accuracy by 8% on the LSTM and an 18% on the CNN.

**AB** Both author profilers achieved high accuracies (ca. 0.90) on the original AB data. Neither A^4NT nor ParChoice managed to decrease accuracy to below 0.50. ParChoice performed better in the Alice→Bob direction and A^4NT in the converse, for both the LSTM and CNN profiler. At most, A^4NT was able to take accuracy down by 30%, and ParChoice by 28%. Imitation performance was thus highly similar between the two techniques in the AB dataset, but somewhat concentrated to opposing directions.

**TO** Unlike with the other datasets, in TO we found a clear difference in performance between the LSTM and CNN profilers, the former performing systematically better on the original test set in both classes. Interestingly, however, A^4NT was able to take the LSTM’s accuracy down much more than the CNN’s, clearly surpassing the imitation limit with the LSTM in both directions. The CNN was far more resistant to A^4NT’s imitations, to the point of its accuracy increasing in the Obama→Trump direction. ParChoice did not surpass the imitation limit in TO, but still systematically took accuracy down with both profilers and directions.

### 7 DISCUSSION

**Interpretation of results** The empirical results presented in Section 6 can be summarized as three main findings. First, the CAE and BT baselines failed to produce acceptable paraphrases in any dataset, as measured both by our own manual evaluations and the

| Dataset | Direction | Author profiler | Profiler accuracy | Profiler accuracy |
|---------|-----------|-----------------|-------------------|-------------------|
|         |           |                 | Original | A^4NT | ParChoice |
| YG      | female→male | LSTM            | 0.83     | 0.64  | 0.46  |
|         |            | CNN             | 0.75     | 0.57  | 0.32  |
|         | male→female| LSTM            | 0.82     | 0.70  | 0.48  |
|         |            | CNN             | 0.88     | 0.73  | 0.59  |
| BG      | female→male| LSTM            | 0.74     | 0.76  | 0.68  |
|         |            | CNN             | 0.77     | 0.80  | 0.62  |
|         | male→female| LSTM            | 0.41     | 0.38  | 0.36  |
|         |            | CNN             | 0.39     | 0.32  | 0.35  |
| AB      | Alice→Bob  | LSTM            | 0.86     | 0.77  | 0.62  |
|         |            | CNN             | 0.92     | 0.80  | 0.64  |
|         | Bob→Alice  | LSTM            | 0.94     | 0.71  | 0.78  |
|         |            | CNN             | 0.93     | 0.63  | 0.78  |
| TO      | Trump→Obama| LSTM            | 0.86     | 0.35  | 0.66  |
|         |            | CNN             | 0.61     | 0.47  | 0.55  |
|         | Obama→Trump| LSTM            | 0.77     | 0.23  | 0.51  |
|         |            | CNN             | 0.67     | 0.68  | 0.58  |

Table 8: Imitation results (lower score is better for imitation, lowest scores emphasized)
user study involving independent evaluators. Since semantic retainment is mandatory for any real-world style imitation, we assert that CAE and BT are not usable for this purpose.

Second, even though A^4NT maintained a superior semantic retainment to the other baselines, this was largely due to its tendency to reproduce sentences rather than its ability to enact semantically acceptable transformations. While A^4NT received better overall results in the user study as CAE and BT, its mean (1.7) and median (1) illustrate the poor quality of imitations in human judgement. Since the user study only included imitations that were non-identical with the original sentence, our results indicate that when A^4NT enacts a transformation, it is highly unlikely to retain the original meaning. Manual evaluation results accord with this conclusion. Discarding identical imitations was likely a major reason why we received much lower scores for A^4NT compared to Shetty et al.'s [55] original user study. ParChoice performed significantly better than A^4 NT in semantic retainment, in both the user study and manual evaluation.

Third, ParChoice outperformed A^4NT in most imitation tasks, and maintained a consistent performance of lowering profiling accuracy. A^4NT, in contrast, resulted in a higher post-imitation accuracy on three test settings. This result is especially important considering that ParChoice has been trained on an independent training set and an unrelated simple surrogate profiler (LR), whereas A^4NT has used an overlapping training set and the corresponding model architecture with the LSTM author profiler.

**Tuning ParChoice** We chose the hyperparameter values reported in Section 5.1.3 based on preliminary tests on the YG dataset. Increasing #iterations from 1 to 10 had a systematic but relatively small positive effect on imitation success (< 10% increase in imitation). Our maximum #candidates value 1000 seemed to be high enough for our purposes, as increasing it to 100000 had no marked effect on imitation. We did not systematically experiment on different values of the maximum edit distance parameter, but lowering it would likely improve semantic retainment at the expense of imitation performance. More experimentation is needed to find optimal hyperparameter values for ParChoice.

## 8 CONCLUSIONS AND FUTURE WORK

We showed that ParChoice significantly improves on the state-of-the-art in style imitation in terms of achieving imitation success while retaining semantic content. However, much room for improvement remains, especially related to improper synonym replacement from WordNet. In future work, we plan on experimenting with more sophisticated word sense disambiguation algorithms to reduce such errors. We will further empirically study the effects of parameter choice in ParChoice. Additionally, applying ParChoice to datasets beyond gender and author profiling, as well as multi-class settings, are important for the future development of this technique.

**ACKNOWLEDGMENTS**

We thank Sam Spilsbury and Andrei Kazlouski for their help in implementation, and Mika Juuti for valuable discussions related to the project. This work is supported in part by the Helsinki Doctoral Education Network in Information and Communications Technology (HICT).

**REFERENCES**

[1] Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information and System Security*, 26(2):1–29, 2008.

[2] Afroza Ahmed and King Ip Lin. Negative sentence generation by using semantic heuristics. In *The 52nd Annual ACM Southeast Conference (ACMSE 2014)*, 2014.
A ALL IMITATION RESULTS

Table 10 presents the imitation results from all style imitation techniques, including CAE and BT. In YG their performance was similar to ParChoice’s; somewhat worse in female→male and slightly better in female→male. In BG they failed to take accuracy down in the female→male direction, behaving similarly to A4NT in this respect. The seemingly best imitation performance is achieved by CAE and BT on the AB dataset, where both are able to take accuracy below 0.20 in both directions. As our semantic retention evaluation demonstrated, this was due to CAE and BT largely repeating words common in the target corpus but unrelated to the original sentence. Finally, both CAE and BT successfully imitated Trump but not Obama in TO, CAE actually taking accuracy higher in the Trump→Obama direction.
| Dataset | Direction | Author profiler | Profiler accuracy |
|---------|-----------|----------------|------------------|
|         |           |                | Original | CAE | BT | A^|NT | ParChoice |
| YG      | female→male | LSTM          | 0.83      | 0.55 | 0.54 | 0.64 | 0.46 |
|         |            | CNN           | 0.75      | 0.44 | 0.42 | 0.57 | 0.32 |
|         | male→female | LSTM          | 0.82      | 0.48 | 0.45 | 0.70 | 0.48 |
|         |            | CNN           | 0.88      | 0.59 | 0.53 | 0.73 | 0.59 |
| BG      | female→male | LSTM          | 0.74      | 0.80 | 0.72 | 0.76 | 0.68 |
|         |            | CNN           | 0.77      | 0.81 | 0.79 | 0.80 | 0.62 |
|         | male→female | LSTM          | 0.41      | 0.29 | 0.36 | 0.38 | 0.36 |
|         |            | CNN           | 0.39      | 0.26 | 0.22 | 0.32 | 0.35 |
| AB      | Alice→Bob  | LSTM          | 0.86      | 0.17 | 0.18 | 0.77 | 0.62 |
|         |            | CNN           | 0.92      | 0.22 | 0.15 | 0.80 | 0.64 |
|         | Bob→Alice  | LSTM          | 0.94      | 0.15 | 0.11 | 0.71 | 0.78 |
|         |            | CNN           | 0.93      | 0.09 | 0.08 | 0.63 | 0.78 |
| TO      | Trump→Obama | LSTM          | 0.86      | 0.95 | 0.64 | 0.35 | 0.66 |
|         |            | CNN           | 0.61      | 0.88 | 0.61 | 0.47 | 0.55 |
|         | Obama→Trump | LSTM          | 0.77      | 0.01 | 0.01 | 0.23 | 0.51 |
|         |            | CNN           | 0.67      | 0.00 | 0.01 | 0.68 | 0.58 |

Table 10: All imitation results