Changing the Narrative Perspective: From Ranking to Prompt-Based Generation of Entity Mentions

Mike Chen[1], Razvan Bunescu[2]

[1]School of Electrical Engineering and Computer Science, Ohio University
[2]Department of Computer Science, University of North Carolina at Charlotte

Abstract

Changing the point of view of a character in a story alters the reading experience by reshaping the reader's involvement and identification with the character's thoughts and feelings. An effective NLP solution to the task of changing the narrative perspective requires the capability to select mention strings that refer to the character in a natural and non-ambiguous manner. In this paper, we introduce and evaluate three mention selection architectures: LSTMs with attention over frozen BERT embeddings, fine-tuned BERT with coreference-modulated self-attention, and prompt-based tuning over either frozen or fine-tuned T5. Experimental evaluations show that the prompt-tuning approach over frozen T5 obtains the best performance, also outperforming the previous state-of-the-art on this task.

Keywords

narrative perspective, mention selection, ranking, prompt-tuning

1. Introduction

The narrative point of view (PoV), or perspective, is the position from which the events in a story are observed and communicated. There are 3 types of narrative perspective: *first*, *second*, and *third person*. In fiction, characters are most commonly developed using the first or third person perspective. When employing the first person mode of storytelling, the narrator is usually a character inside the story, recounting events from their own point of view. Conversely, in the third person point of view, the narrator places themselves outside the events in the story. The second person point of view is more common in poetry, how-to guides, technical writing, and self-help texts. In this type of narrative perspective, the reader becomes a character who is addressed by the writer using second person pronouns. Motivated by potential style-transfer applications in fiction writing, in [1] we introduced the task of changing the narrative perspective and described an end-to-end NLP pipeline for shifting the PoV from deictic (1st or 2nd person) to anaphoric (3rd person), as shown in the example below taken from [2]:

For a second, $[I]_1$ → $[\textbf{Jean-Paul}]_1$ *had the idea of getting up, slapping the* $[\textit{Self-Taught Man}]_2$ *on the shoulder and starting a conversation with* $[\textit{him}]_2$*. But just at that moment* $[\textit{he}]_2$ → $[\textbf{the man}]_2$ *caught* $[\textit{my}]_1$ → $[\textbf{his}]_1$ *look.*

Table 1, reproduced from [1], shows potential applications of the PoV change task in the

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): Proceedings of the Text2Story'22 Workshop, Stavanger (Norway), 10-April-2022

mc277509@ohio.edu (M. Chen); razvan.bunescu@uncc.edu (R. Bunescu)

Table 1
Pairs of sentences written from different narrative perspectives.

	Autobiography ↔ Biography
1st	When the youngest of *my* sisters saw him in his bloodied condition, *I* clearly recall her bursting out ”Let me die in his place!”. [4]
3rd	When the youngest of *Akira's* sisters saw her brother in his bloodied condition, *Akira* clearly recalls her bursting out ”Let me die in his place!”.

	Educational ↔ Self-diagnosis
3rd	*Persons afflicted with the classic form of the illness* have periods of severe depression as well as periods of mania. [5]
2nd	*You* have periods of severe depression as well as periods of mania.

	Abstract ↔ Related work
1st	*We* show that the bound is tight, and *we* develop an algorithm that computes the optimal schedule. [6]
3rd	*The authors* show that the bound is tight, and *they* develop an algorithm that computes the optimal schedule.

automatic generation of biographical text, self-diagnosis material, or related work paragraphs, starting from auto-biographical text, educational material, or abstract paragraphs, respectively. While changing the narrative perspective bears similarities with other NLP tasks, such as paraphrasing, referential expression generation, and style transfer, it has unique aspects that require customized, if not entirely novel solutions [1].

Following the end-to-end pipeline described in [1], a coreference resolution system is used first to identify the entity mentions for all characters in the story. The example above contains two discourse entities: the narrator *Jean-Paul*, using a 1st person PoV, and *the Self-Taught Man*, using a 3rd person PoV. Given a *focus entity E = Jean-Paul* that is described in the 1st person PoV, the task is to change its mention strings to reflect a 3rd person PoV. This is done by a *mention selection* model that is tasked with choosing among strings in a set $S(E)$ that contains names, nouns, as well as suitable 3rd person pronouns. This set of candidate strings is created automatically by a separate component of the pipeline. Another pipeline module is tasked with changing verb conjugations from 1st to 3rd person whenever the focus mention is the subject of a verb at present tense singular. The mention selection task is non-trivial, as there may already be other *confounding entities*, e.g. *the Self-Taught Man*, that are mentioned using the same third person pronouns, as such their mention strings might need to be changed too, in order to avoid referential ambiguity. At the same time, repeated uses of names should be avoided, to maintain naturalness. In [1] we describe a ranking approach to mention selection that processes the text auto-regressively using LSTMs on top of BERT embeddings [3].

In this paper, we show that adding self-attention to the LSTM architecture (Section 2.1) improves both mention selection and end-to-end performance. Furthermore, we introduce two new architectures: a coreference-augmented self-attention model for BERT (Section 2.2) that eliminates the LSTM layer, and a prompt-tuning approach (Section 3) for the T5 text-to-text Transformer [7]. Overall, prompt-tuning [8] over T5 with frozen parameters performs the best, with further gains observed when fine-tuned on the PoV dataset in a 2-fold evaluation.

2. Mention Selection as Ranking

Given a set of candidate strings $S(E)$ that can be used for referring to an entity E, mention selection is about determining the most appropriate string to use in a given textual context C. For a confounding entity, this set is determined from all unique strings that are used in its coreference chain, e.g. $S(E_2)$ = {*the Self-Taught Man*, *the man*, *he*, *his*, ...} in the example above. For the focus entity, which originally is in a 1st or 2nd person PoV, we use 3rd person pronouns that agree in number and gender with the given name, as well as noun phrases extracted from the document using the methods described in [1]. A scoring function $score(s_j|C)$ is trained to capture how appropriate a string $s_j \in S(E)$ is to be used as a mention of an entity E in context C. Given $s_i, s_j \in S(E)$, we use $\langle s_i \succ s_j|C\rangle$ to denote that s_i is more appropriate than s_j in the context C. Correspondingly, the ranking system is trained to compute a $score(s_i|C) > score(s_j|C) + \gamma$, where γ is a margin hyper-parameter, which results in the margin-based ranking loss shown below:

$$\sum_{s_i \succ s_j|C} max\{0, \gamma - score(s_i|C) + score(s_j|C)\}$$

At training time, we use the observed mention string $\hat{s} \in S(E)$ to create ranking pairs $\langle \hat{s} \succ s|C\rangle$, for all $s \in S(E), s \neq \hat{s}$, as training examples.

2.1. LSTMs over Tokens and Mentions

The best performing approach in [1] is composed of 4 LSTM models. For any given entity mention, the text is split into a left context C_{left} that ends with the current mention and a right context C_{right}, i.e. $C = [C_{left}, C_{right}]$. Mention strings in C_{right} corresponding to focus and confounding entities are replaced with a special $\langle unk\rangle$ token, to ensure models do not use information unavailable at test time. The left context is processed sequentially, at token-level by $LSTM^T_{left}$ and at mention-level by $LSTM^M_{left}$, producing the final states h^T_{left} and h^M_{left}, respectively. A similar processing is done for the right context, to produce final states h^T_{right} and h^M_{right}. All LSTMs are run on top of contextual embeddings produced by a frozen BERT [3]. The final states are then concatenated and used as input to a fully connected network with one hidden layer and a linear output node that computes the $score(s_j|C)$.

Given the significant gains in performance brought about by attention when used with LSTMs [9] or within Transformer [10], we added an attention mechanism to each of the four LSTM models. Using the concatenation formulation of Luong et al. [11], we compute a context vector z for the last token of the current mention, where the attention weights span all the tokens to the left or to the right of the current mention, depending on whether the left or right LSTM is used. For the token-level LSTMs, the two context vectors are concatenated and used as input to a fully connected network with one hidden layer to compute a $score(z^T_{left}, z^T_{right})$. Similarly, a $score(z^M_{left}, z^M_{right})$ is computed using the two context vectors from the mention-level LSTMs. Finally, the two attention-based scores are added to the original LSTM-based ranking score described above in order to compute the final ranking $score(s_j|C)$. The overall architecture containing the original LSTMs and the new attention mechanisms is illustrated in Figure 1.

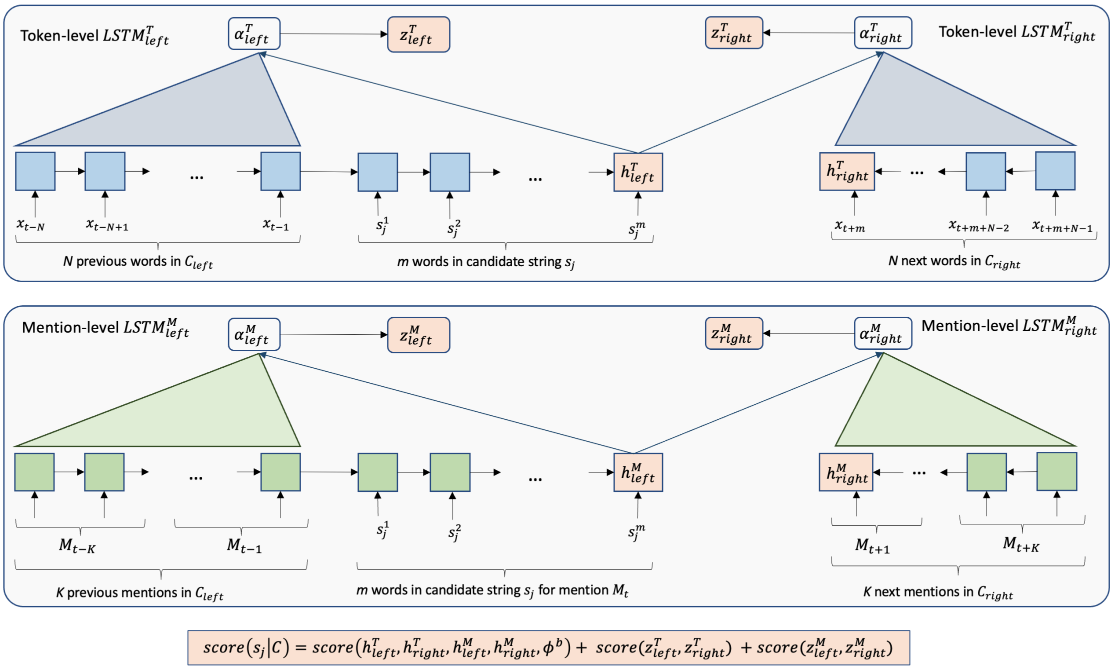

Token-level $LSTM^T_{left}$
α^T_{left}
z^T_{left}
z^T_{right}
α^T_{right}
Token-level $LSTM^T_{right}$
h^T_{left}
h^T_{right}
x_{t-N}
x_{t-N+1}
x_{t-1}
s_j^1
s_j^2
s_j^m
x_{t+m}
$x_{t+m+N-2}$
$x_{t+m+N-1}$
N previous words in C_{left}
m words in candidate string s_j
N next words in C_{right}
Mention-level $LSTM^M_{left}$
α^M_{left}
z^M_{left}
z^M_{right}
α^M_{right}
Mention-level $LSTM^M_{right}$
h^M_{left}
h^M_{right}
M_{t-K}
M_{t-1}
M_{t+1}
M_{t+K}
K previous mentions in C_{left}
m words in candidate string s_j for mention M_t
K next mentions in C_{right}
$score(s_j|C) = score(h^T_{left}, h^T_{right}, h^M_{left}, h^M_{right}, \phi^b) + score(z^T_{left}, z^T_{right}) + score(z^M_{left}, z^M_{right})$

Figure 1: Computation of $score(s_j|C)$ using token-level and mention-level LSTMs with attention for left and right contexts, and the vector of binary features ϕ^b introduced in [1].

2.2. Coreference-Modulated Self-Attention

The LSTM-based architecture uses as input contextual embeddings computed by BERT. Inspired by the concept of *pseudo self-attention* [12], we developed a new approach that does away with the LSTM layers, and instead adapts the BERT model itself to compute a vector representation of the candidate mention string s_j in context. To incorporate coreference information in the underlying Transformer model, we first introduce a square matrix $Y \in \mathbb{Z}^{N \times N}$ that represents the coreference information in the N input tokens, where $Y[i, j] = 1$ if and only if the tokens at positions i and j belong to mentions that are coreferent, otherwise $Y[i, j] = 0$. Correspondingly, the diagonal vector $\mathbf{d}$ of the coreference matrix Y has $d[i] = 1$ if the token i belongs to a person entity mention, otherwise $d[i] = 0$. If we update each row in Y as $Y[i, :] = 2 \times \mathbf{d} - Y[i, :]$, then $Y[i, j] = 0$ if the tokens at positions i and j do not belong to any entity mentions; $Y[i, j] = 1$ if the two tokens belong to mentions that are coreferent; and $Y[i, j] = 2$ if token j belongs to an entity mention that does not corefer with the entity mention to which token i belongs, if any. Thus, for each token position i, the corresponding row $Y[i, :]$ will contain one of the numbers 0, 1, and 2, distinguishing among the 3 situations. We map each of the 3 numbers to their own trainable embedding with size d_{model}, and then transform the row vector $Y[i, :]$ into a coreference embedding matrix $C[i] \in \mathbb{R}^{N \times d_{model}}$ by replacing the numbers 0, 1, and 2 with their corresponding embeddings. When the matrices $C[i]$ are stacked over all token positions i in the input, they create a 3-dimensional coreference embedding tensor $C \in \mathbb{R}^{N \times N \times d_{model}}$. Let $P \in \mathbb{R}^{N \times N \times d_{model}}$ be the 3-dimensional tensor of relative positional embeddings [13]. Then, for each attention layer in BERT, we define an additional attention mechanism where the unnormalized attention weights e_{ij} are computed using the input embedding $\mathbf{x}_j$ at that layer, the coreference embedding $C[i, j]$ and the relative positional embedding $P[i, j]$ between positions i and j. The corresponding vectorization is then done using an Einstein summation operator $\star$ as shown below, where the input embedding matrix X is broadcast over the first dimension of C:

$$e_{ij} = \frac{((\mathbf{x}_j + C[i, j])U^Q)(P[i, j]U^K)^T}{\sqrt{d_k}}; \qquad \text{softmax}\left(\frac{((X + C)U^Q) \star (PU^K)^T}{\sqrt{d_k}}\right)(VU^V)$$

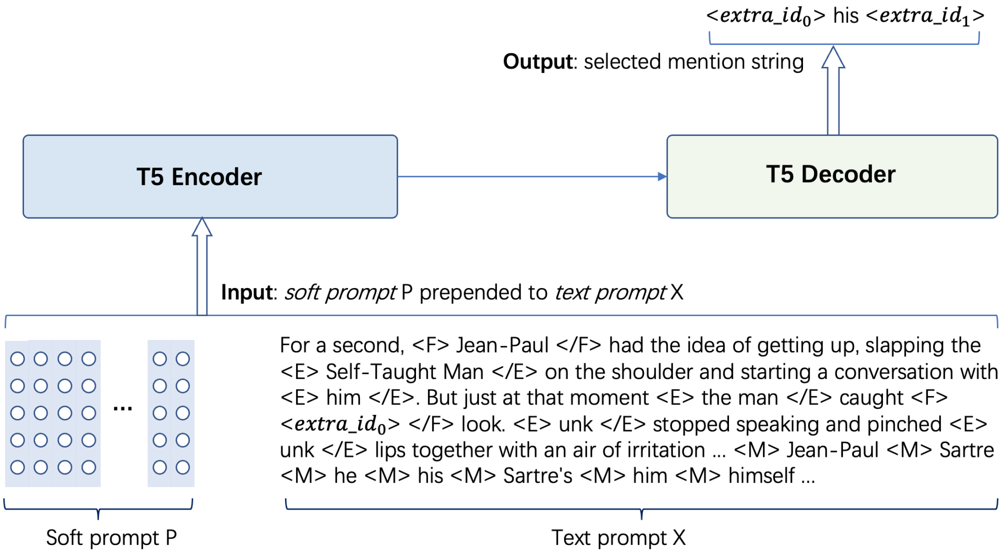
At each layer, the coreference-based embeddings computed above are added to the original self-attention embeddings computed by the frozen BERT model, resulting in a layer of *coreference-modulated* embeddings. The new set of parameters U^K, U^Q, and U^V, which mirror the original BERT parameters W^K, W^Q, and W^V, are instantiated and trained for each layer in the Transformer. The coreference-modulated embedding computed in the final layer for the last token of the current mention s_j is concatenated with the binary features ϕ^b introduced in [1] and used as input for a fully connected network with one hidden layer, followed by a linear node that outputs the final ranking $score(s_j|C)$.

3. Mention Selection as Prompt-Based Generation

The coreference-modulated self attention approach, while keeping the original BERT parameters frozen, still requires training the new sets of attention parameter matrices U^Q, U^K, U^V for each Transformer block. For the particular BERT model used in the experiments, this means over 8.5 new million new parameters will need to be trained from scratch, which is time consuming and liable to lead to overfitting. In this section we describe an alternative, much less parameter-intensive method of utilizing coreference information, based on *P*-tuning*. This general class of techniques subsumes methods such as soft prompt-tuning [8], p-tuning [14], and prefix-tuning [15] in which the encoder of the language model (LM) is run on an input composed of two parts: (1) the usual textual input, possibly augmented with additional tokens that convey information about the task; and (2) a set of token embeddings that are trained from scratch. These continuous embeddings, also called *soft tokens*, do not have to correspond to actual language tokens and are meant to help the LM adapt its ouput for the target task. Of the three P*-tuning approaches listed above we chose to use the soft prompt-tuning approach [8], which is relatively simpler. For the LM, we use the T5 text-to-text Transformer of Raffel et al. [7] in the span corruption + sentinel mode. Figure 2 illustrates the prompt-based approach where a frozen T5 is run on an input that contains two prompts:

1. A soft prompt P composed of 20 continuous embeddings that are initialized with random token embeddings from the T5 vocabulary.
2. A text prompt X composed of the left and right context words around the current entity mention, augmented with the strings in $S(E)$.

In order for the T5 model to know what the candidate mention strings are, the strings in $S(E)$ are included in the text prompt at the end, separated by $\langle M \rangle$ tags. The current mention is indicated



<*extra_id*$_0$> his <*extra_id*$_1$>
Output: selected mention string
T5 Encoder
T5 Decoder
Input: *soft prompt* P prepended to *text prompt* X
...
For a second, <F> Jean-Paul </F> had the idea of getting up, slapping the <E> Self-Taught Man </E> on the shoulder and starting a conversation with <E> him </E>. But just at that moment <E> the man </E> caught <F> <*extra_id*$_0$> </F> look. <E> unk </E> stopped speaking and pinched <E> unk </E> lips together with an air of irritation ... <M> Jean-Paul <M> Sartre <M> he <M> his <M> Sartre's <M> him <M> himself ...
Soft prompt P
Text prompt X

Figure 2: Architecture of the prompt-based model for mention generation.

by the sentinel token $\langle extra\_id_0 \rangle$, which during T5's pre-training was used to indicate the span of text that needs to be generated by the decoder. Furthermore, focus mentions are delimited by $\langle F \rangle$ and $\langle /F \rangle$ tags, whereas confounding mentions are enclosed between $\langle E \rangle$ and $\langle /E \rangle$ tags. The $\langle unk \rangle$ tag is used to specify unknown strings for future mentions, i.e. mentions of the focus or confounding entities that appear to the right of the current mention. An example prompt is shown in Figure 2. The decoder is then tasked with generating a sequence Y containing the correct mention string *his*, formatted using the T5 sentinels. While T5's parameter are kept fixed, we expect the tuning of the soft-prompt to enable it to learn to generate the correct mention string by copying it from the input text prompt sequence.

The resulting prompt-based model is trained under two scenarios: (1) *prompt-tuning with pre-trained T5* and (2) *prompt-tuning with fine-tuned T5*. In the first scenario, the T5 parameters are frozen and the only parameters that are updated are the soft-prompt embeddings in P and the embeddings for the special tags introduced above. In the second scenario, the T5 parameters are allowed to change too during backpropagation of the loss.

4. Experimental Results and Error Analysis

All mention selection models are trained on the training portion of the CoNLL-2012 dataset [16]. For each document in the corpus, we use one 3rd person coreference chain at a time as a focus entity, which is assumed to have been transformed from 1st to 3rd person PoV, whereas the remaining chains that agree in number and gender are used as confounding entities. Early stopping and hyper-parameter tuning are done on the development portion of CoNLL. The trained models are then evaluated *within-distribution* for mention selection on the test portion of CoNLL, and *out-of-distribution* on the PoV dataset introduced in [1] that consists of 21 documents covering a wide array of types of narratives, where 300 entities are mentioned 8,682

Table 2
Comparative performance (%) in terms of Accuracy for Mention Selection, Precision (P), Recall (R), and F1 for End-to-End (E2E) PoV Change. LSTM over BERT refers to the best model from [1]; LSTM with Attention over BERT uses the attention mechanism described in Section 2.1; BERT with Coreference Self-Attention is described in Section 2.2; The prompt-tuning approaches are described in Section 3.

	Mention Selection		PoV Change E2E		
Model	CoNLL	PoV	P	R	F1
LSTM over BERT	75.9	74.9	70.1	76.2	73.1
LSTM with Attention over BERT	77.1	75.7	70.9	77.0	73.9
BERT with Coreference Self-Attention	77.4	75.7	69.7	75.8	72.7
Prompt-tuning with pre-trained T5	78.4	78.5	72.7	79.0	75.7
+ 2-fold fine-tuning on PoV	75.5	82.6	**74.2**	**80.6**	**77.3**
Prompt-tuning with fine-tuned T5	77.1	77.4	68.9	74.8	71.8
+ 2-fold fine-tuning on PoV	74.7	81.8	73.2	79.5	76.2

times in total. To evaluate end-to-end performance on changing the narrative perspective, we plug each mention selection model into the PoV change pipeline that also performs coreference resolution, extraction of candidate mention strings $S(E)$, syntactic parsing, and verb conjugation change. Additionally, for each prompt-tuning model we also evaluate its within-distribution performance on the PoV dataset. This is done in a 2-fold evaluation scenario where the PoV dataset is first partitioned at random into 2 folds: fold_1 containing 11 documents and fold_2 containing the remaining 10 documents. In the first evaluation step, the prompt-based models that were trained on CoNLL are further fine-tuned on fold_1 and tested on fold_2; in the second step, the roles of the two folds are swapped, and the prompt-based models are fine-tuned on fold_2 and tested on fold_1. The test results are then pooled over the 2 folds in order to compute the overall within-distribution performance on the PoV dataset.

The overall results are shown in Table 2, using accuracy for mention selection and precision (P), recall (R), and F1-measure (F1) for the end-to-end performance on the PoV dataset. The results show that adding attention to the original LSTM model improves performance across all evaluations. The coreference-augmented self-attention model, while matching LSTMs with attention on CoNLL, is under-performing when tested on the PoV dataset, which could be explained by overfitting to CoNLL. The best performance in terms of out-of-distribution generalization to the PoV dataset is obtained by prompt-tuning using the frozen T5 model, with an F1 measure of 75.7%. When fine-tuned on the PoV dataset in the 2-fold evaluation setting, F1 measure is further increased to 77.3%. Compared to the other approaches, prompt-tuning is overall simpler, is faster at training due to the much smaller number of trainable parameters, and does not use engineered features, i.e. the binary features ϕ^b used in the other models.

It is important to note here that the results are likely to be much better when the system outputs are evaluated by human readers, as there may be multiple good solutions for choosing mention strings that achieve felicitous, non-ambiguous reference while also maintaining the naturalness of a text. This was verified in [1] for the LSTM over BERT model, where Amazon Mechanical Turk workers were observed to give *referential* and *naturalness* scores to the system output that were not very far from the scores given to manual annotations. Indeed, upon doing error analysis on the output of the prompt-tuning model, we found many instances like the ones below, where the mention string chosen by the model (shown in light red) was comparable in naturalness and referential clarity with the annotated string (shown in light gray):

1. *It was eight in the morning and* [*Katz*]$_1$ *looked very happy.* [*He*]$_1$ *was always happy when* [*he*]$_1$ *was drunk, and* [*he*]$_1$ *was always drunk. Two weeks after that,* [*I*]$_2$ → **[Bill]**$_2$ *later heard, police found* [*him*]$_1$ → **[Katz]**$_1$ *in an upended car in a field outside the little town of Mingo, hanging upside down by his seatbelt.*
2. [*I*]$_2$ → **[Bill]**$_2$ *found* [*myself*]$_2$ → **[himself]**$_2$*, six days later, standing at* [*our*]$_4$ → **[their]**$_4$ *local airport watching a tin commuter plane containing* [*Katz*]$_1$ *touch down ... For the past three years* [*Katz*]$_1$ *had devoted* [*himself*]$_1$ *to rectitude and –* [*I*]$_2$ → **[Bill]**$_2$ *instantly saw now as* [*he*]$_1$ *stooped out the door of the plane – growing a stomach.* [*Katz*]$_1$ → **[He]**$_1$ *was arrestingly larger than when* [*I*]$_2$ → **[Bill]**$_2$ *had last seen* [*him*]$_1$.
3. [*Both boys*]$_3$ *had closed* [*their*]$_3$ *dictionaries.* [*The brown haired one*]$_1$ → **[The brown haired boy]**$_1$ *was not talking,* [*his*]$_1$ *face, stamped with deference and interest, ...*
4. *As it had for many of the guides* [*I*]$_1$ → **[Michael]**$_1$ *had met, the mystical experience* [*Fritz*]$_2$ *had on psychedelics launched* [*him*]$_2$ *on a decades long spiritual quest that eventually "blew my linear, empirical mind", opening* [*him*]$_2$ *up to the possibility of past lives, telepathy, precognition, and "synchronicities" that defy our conceptions of space and time.* [*He*]$_2$ → **[The man]**$_2$ *spent time on an ashram in India, where* [*he*]$_2$ *witnessed specific scenes that had been prefigured in* [*his*]$_2$ *psychedelic journeys.*

In the first example, the system generates proper names instead of pronouns, which improves referential clarity, with perhaps a slight decrease in naturalness. The second example illustrates the opposite behavior, where the system generates a pronoun instead of the manually annotated proper name, which actually makes the text sound more natural while maintaining referential clarity. In the third example, the noun phrase chosen by the system is equally good in terms of referential clarity, albeit its syntactic head "boy" may sound repetitive and hence less natural than the annotated "one". In the last example, instead of a pronoun, the system selects a nominal mentions string, which appears to be as appropriate in the context as the pronoun.

5. Conclusion

We introduced new mention selection models targeted at the task of changing the narrative perspective from deictic (1st or 2nd person) to anaphoric (3rd person). Adding an attention mechanism to a previous state-of-the-art LSTM model that is trained on top of frozen BERT embeddings was shown to improve its performance. We also introduced a new BERT model with coreference-modulated self-attention, and a soft prompt-tuning approach for the T5 text-to-text Transformer, with the later shown to significantly improve both the within- and out-of-distribution generalization performance. Code, hyper-parameter settings, and pre-trained models are made publicly available[1]. More general models that can also modify the text between entity mentions are planned for future work.

[1] https://github.com/chenmike1986/change_pov

References

[1] M. Chen, R. Bunescu, Changing the narrative perspective: From deictic to anaphoric point of view, Information Processing & Management 58 (2021) 102559. doi:https://doi.org/10.1016/j.ipm.2021.102559.

[2] J.-P. Sartre, Nausea, New Directions, 1969.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL 2019, ACL, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[4] A. Kurosawa, Akira Kurosawa: Something Like an Autobiography, Vintage, 1983.

[5] F. M. Mondimore, Bipolar Disorder: A Guide for Patients and Families, John Hopkins University Press, 2006.

[6] N. G. Gast, D.-C. Tomozei, J.-Y. Le Boudec, Optimal generation and storage scheduling in the presence of renewable forecast uncertainties (2013) 11. URL: http://infoscience.epfl.ch/record/183046.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.

[8] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: Proceedings of EMNLP 2021, ACL, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. doi:10.18653/v1/2021.emnlp-main.243.

[9] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR, 2015, pp. 1–15.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[11] T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of EMNLP 2015, ACL, Lisbon, Portugal, 2015.

[12] Z. M. Ziegler, L. Melas-Kyriazi, S. Gehrmann, A. M. Rush, Encoder-agnostic adaptation for conditional language generation, 2019. arXiv:1908.06938.

[13] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, 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, New Orleans, Louisiana, 2018.

[14] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021.

[15] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, GPT understands, too, 2021. arXiv:2103.10385.

[16] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, Y. Zhang, CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes, in: Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012), Jeju, Korea, 2012.