"Politeness, you simpleton!" retorted [MASK]: Masked prediction of literary characters

Eric Holgate
The University of Texas at Austin
Department of Linguistics
holgate@utexas.edu

Katrin Erk
The University of Texas at Austin
Department of Linguistics
katrin.erk@utexas.edu

Abstract

What is the best way to learn embeddings for entities, and what can be learned from them? We consider this question for the case of literary characters. We address the highly challenging task of guessing, from a sentence in the novel, which character is being talked about, and we probe the embeddings to see what information they encode about their literary characters. We find that when continuously trained, entity embeddings do well at the masked entity prediction task, and that they encode considerable information about the traits and characteristics of the entities.

1 Introduction

Neural language models have led to huge improvements across many tasks in the last few years (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019).[1] They compute embeddings for words and word pieces. But when we describe the semantics of a sentence, we talk about entities and events and their relations, not words. And it is to be expected that more complex reasoning tasks would eventually require representations at the semantic level rather than the word level. Entities differ from words in that they are persistent, localized, and variable (within a given range). So, would it be beneficial to compute embeddings of entities in addition to embeddings of words for downstream inference? And how should entity embeddings be computed?

There has been a steady rise in work on entity representations and how they can be combined with language models, for example Li et al. (2016); Bosselut et al. (2018); Rashkin et al. (2018); Louis and Sutton (2018).In this paper, we add to the growing literature on neural representations of entities by considering a particularly challenging case: the representations of entities in very long texts, in particular in novels. Intriguingly, Bruera (2019) recently tested whether literary characters, when represented through distributional vectors trained on the first half of a novel, can be recognized in the second half, and found the task to be near impossible. We take up that same task, but train character embeddings in a masked character prediction task. We ask the following questions. (a) Is it possible to use literary character embeddings to do masked character prediction, that is, to guess from a sentence in a novel which character it mentions? (b) If this task is doable, is it doable only locally, or can we train on the first third of a novel and then guess characters towards the end of the novel? (c) What do the resulting embeddings tell us about the literary characters when we probe them? (d) Can the embeddings identify a literary character from a short description of their personality?

We find that when continuously trained, entity embeddings do well at the masked entity prediction task, and that they encode considerable information about the traits and characteristics of the entities. Modeling semantics for natural language understanding is about modeling entities and events, not words. So we view this work as an initial step in the direction of entity modeling over time.

2 Related Work

Entities have been increasingly common subjects within NLP research. There has been recent work aimed at inducing both characteristics of entities, such as personalities, physical and mental states, and character traits, as well as distributed entity representations, similar to lexical embeddings.

[1] The [MASK] in the title is actually La Carconte, from the *Count of Monte Cristo* by Alexandre Dumas.

Proceedings of the 14th International Conference on Computational Semantics, pages 202–211
June 17–18, 2021.

2.1 Modeling Personalities and Characteristics

Psychologists have studied the relationship between *personality traits* and *human behavior*. Within NLP, there have been recent attempts to model this link computationally.

Bamman et al. (2013) explored entity modeling by using Bayesian methods to induce moderately fine-grained character archetypes/stereotypes from film plot summaries. The authors utilized dependency relations to identify, for each entity, the verbs for which they were the agent, the verbs for which they were the patient, and any modifiers attributed to them. Bamman et al. successfully induced clusters that could be manually aligned with tropes like *the jerk jock*, *the nerdy klutz*, *the villain*, etc.

Plank and Hovy (2015) recently appealed to psychological personality dimensions in relation to linguistic behavior. They constructed a dataset by crawling twitter for mentions of any of the 16 Myers-Briggs Type Indicators comprising four personality dimensions (MBTI; Myers and Myers 2010), labeling tweets with author gender identity. Plank and Hovy then train logistic regression models to predict each of the four dimensions from user tweet data using tweet context features and other features that are traditional for Twitter data (e.g., counts of tweets, followers, favorites, etc.). In all four dimensions, logistic regression classifiers outperform majority baselines, supporting the notion that linguistic behavior correlates with MBTI designations.

Flekova and Gurevych (2015) similarly explored personality traits, though they utilized the Five-Factor Model of personality instead of MBTIs (John et al., 1999). Here, authors collected extraversion/intraversion ratings for approximately 300 literary characters, and explore three sources of signal to predict the extraversion scores. The first system aligns most closely with Plank and Hovy's work as it considers only character speech (both style and content). Flekova and Gurevych go slightly farther, however, as they also show that character actions and behaviors as well as the descriptions of characters given in narration carry useful signal for extraversion prediction.

Rashkin et al. (2018) modeled the mental state of characters in short stories, including motivations for behaviors and emotional reactions to events. The authors noted a substantial increase in performance in mental state classification when entity-specific contextual information was presented to the classifier, suggesting that entity-specific context may be useful to a wide array of downstream tasks.

Louis and Sutton (2018) further explored the relation between character properties and actions taken in online role-playing game data. In *Dungeons and Dragons*, a giant is more likely than a fairy to wield a giant axe, but a fairy is more likely to be agile or cast spells. Louis and Sutton show that computational models can capture this interaction by using character description information in conjunction with action descriptions to train action and character language models. When a formal representation of a given character is included, performance improves.

Bosselut et al. (2018) demonstrated dynamically tracked cooking ingredients, identifying which ingredient entity was selected in any given recipe step, and recognizing what changes in state they underwent as a result of the action described in the step. For example, these dynamic entity representations enabled the model to determine that an ingredient was clean after having been washed.

2.2 Entity Representations and Entity Libraries

Recently, major work in NLP has begun to explicitly model entities for use in downstream tasks. While still new (and limited in scope), much of this work has relied upon the notion of an Entity Library, a vocabulary of individuals which utilizes consecutive mentions to construct distributed vector representations, though methods of learning these representations have varied.

Entity representations have been shown to improve the quality of generated text. In Ji et al. (2017), researchers build a generative language model (an RNN) which has access to an entity library which contains continuous, dynamic representations of each entity mentioned in the text. The result is that the library explicitly groups coreferential mentions, and each generated mention affects the subsequently generated text.

Tracking entity information has also been shown to be useful for increasing the consistency of responses in dialogue agents (Li et al., 2016). Researchers introduce a conversation model which maintains a persona, defined as the character that the artificial agent performs during conversational interactions. The persona maintains elements of identity such as background facts, linguistic behav-

ior (dialect), and interaction style (personality) in continuously updated distributed representations. The model maintains the capability for the persona to be adaptive, as the agent may need to present different characteristics to different interlocutors as interactions take place, but reduces the likelihood of the model providing contradictory information (i.e., maintaining these distributed representations prevents the model from claiming to live in both Los Angeles, Madrid, and England in consecutive queries). Crucially, this desirable change is achieved without the need for a structured ontology of properties, but instead through persona embeddings that are learned jointly with word representations.

Fevry et al. (2020) demonstrates that entity representations trained only from text can capture more declarative knowledge about those entities than a similarly sized BERT. Researchers showed that these representations are useful for a variety of downstream tasks, including open domain question answering, relation extraction, entity typing, and generalized knowledge tasks.

Yamada et al. (2020) explore another entity masking task in the context of transformer pretraining. They train a large transformer on both masked words and masked entities in Wikipedia text. Here, however, each entity-in-context exists as its own token, rather than a representation that is aggregated over a sequence of mentions. Yamada et al. test on entity typing, relation classification, and named entity recognition.

Finally, Bruera (2019) introduces the data that we will use to build our model (described in detail below), and compares the ability to construct computational embeddings for proper names with that of common nouns. Researchers trained a distributional semantics model to create and store two different representations for literary characters in novels, each from a separate section of text from the novel. The model is then asked to match the characters' representations from one portion of text to the representations computed from the other portion of text, which the authors term the Doppelgänger Task. Importantly, their results showed that the ability to match these representations is much reduced in the case of proper names when compared to common nouns. This insight serves as a major motivation for the current work, where we follow the hypothesis that entities can be represented in a distributional fashion after all, though not with the same training as with common nouns.[2] We assume that entity representations must be persistent, continuously available, and dynamic.

Dataset	Min	Max	Mean
OriginalNovels	6,770	568,531	118,184.7
WikiNovels	484	13,261	5,104.6

Table 1: Document length statistics for each Novel Aficionados dataset.

3 Data

In the current paper, we present a model that is able to construct entity representations for characters in classic literary novels. Novels are a compelling environment for this exploration as they feature a relatively small number of entities that appear frequently over a long document. To this end, we turn to the Novel Aficionados dataset introduced by Bruera (2019).

The dataset comprises 62 pieces of classic literature, represented as both their original texts (deemed the *OriginalNovels dataset*; these texts are distributed by Project Gutenberg, which maintains a repository of free eBooks of works no longer protected by copyright), and their English Wikipedia summaries (the *WikiNovels dataset*). In order to have sufficient description of as many characters as possible, we only utilize the corpus of original novels in training our representations, as this corpus yields significantly more mentions per character. We utilize the Wikipedia summaries as a test set to determine how well our entity representations work outside the domain of the novels themselves.

The novels are distributed within the dataset in both their original form and having been preprocessed with BookNLP (Bamman et al., 2014). BookNLP is a natural language processing pipeline that extends from Stanford CoreNLP (Manning et al., 2014) and is specifically aimed at scaling to books and other long documents. BookNLP includes part of speech tagging, dependency parsing, NER, and supersense tagging. Most critical to our application, BookNLP provides quotation speaker identification, pronominal coreference resolution,[3]

[2] While our work is inspired by Bruera (2019) and conducted on the same data, we introduce a different task that is not directly comparable to the Doppelgänger Task.

[3] Unfortunately, the texts are not distributed with more general coreference resolution (outside of character aliases and pronominal resolution). This means we are unable to include nominal expressions as character mentions to be considered

and character name clustering. This means that, in addition to standard anaphoric coreference resolution, BookNLP can identify different proper names as character aliases (i.e., Jane Fairfax, a character from Jane Austen's *Emma*, is referenced throughout the text not only by her full name, but also *Jane*, *Miss Jane Fairfax*, and *Miss Fairfax*; BookNLP is able to recognize this and map all of these aliases to a single, unique character ID). Concerning the quality of the coreference resolution, Bamman et al. report average accuracy of 82.7% in a 10-fold cross-validation experiment on predicting the nearest antecedent for a pronominal anaphor. While the accuracy of character clustering was not evaluated, manual inspection of the data revealed it to be very reliable.

4 Modeling

Our hypothesis is that it is possible to represent characters in a novel through an embedding in such a way that it is possible for a model to recognize who is who, or, as we call the task here, *Is this Me?*. Bruera (2019) found that with an approach that treated characters like common nouns, the related *Doppelgänger Task* was not feasible.[4] We hypothesize that if embeddings are learned to best facilitate *Is this Me?* prediction, the task will be feasible. We further hypothesize that the resulting embeddings can be found to contain information about the characters. In a way, our approach is similar to recent contextualized language models like BERT (Devlin et al., 2019) in that we, too, train on a masked prediction task, and we, too, hope to find the resulting embeddings to be useful beyond the prediction task itself.

4.1 A model for the "Is this Me?" task

Our model keeps track of characters as they appear in a novel, and trains an embedding for each character through the *Is this Me?* task, a masked prediction task: Given a sentence of the novel with a masked character mention, and given the current embedding for character c, a classifier decides whether this is a mention of c or not. This is a binary task. The embedding for each character is updated incrementally as the novel is read, and as such, the entity embeddings are learned directly from the data. The classifier weights are updated alongside the character embeddings.

Because the classifier weights are learned as the model reads the novels, we read all novels in parallel. The classifier is trained on a binary masked prediction task, where negative examples are drawn from the same novel. (That is, a negative example for Emma in the novel *Emma* might be Harriet, but it would never be Heathcliff.) A sketch of the model is shown in Figure 1.

Entity Library. The entity library, shown in blue in Figure 1 is a collection of embeddings of literary characters, each represented by a 300 dimensional embedding learned incrementally throughout the novel. Entity embeddings are randomly initialized and passed through a projection layer (green in the figure) before being received by the classifier.

Contextual Sentence and Target Mention Representations. We utilize the base, uncased distribution of BERT to compute *contextualized sentence representations* of each target sentence, shown in orange in Figure 1. Contextualized sentence representations are truncated to a maximum of 150 subword tokens.[5] We do not fine tune BERT on our data. All character mentions in a sentence are masked. The input to the classifier is a *target representation* of one of the masked entity mentions, using *mix* representations introduced in (Tenney et al., 2019). The *target mention representation* is computed directly from the contextualized sentence representations obtained from BERT and is a scalar mix of the layer activations using learned scalars.

***Is this Me?* Binary Classifier.** The classifier for the binary *Is this Me?* task takes as input an entity embedding, transformed through the projection layer, along with a target mention embedding from BERT as described above. The classifier consists of a single, ReLU activation layer. We keep the classifier this simple intentionally, as, to be successful, we want the entity representations to do the heavy lifting.

4.2 Model details

Training Data. We restrict our modeling to characters that appear at least 10 times to ensure that

by the model.

[4] Although related, the *Is this Me?* and *Doppelgänger* tasks are truly different in nature. As such, we cannot compare results on the *Is this Me?* task to results on the *Doppelgänger Task* directly.

[5] This limit was determined by inspecting the length of each sentence in the corpus in subword tokens and permits nearly all sentences to remain untruncated.

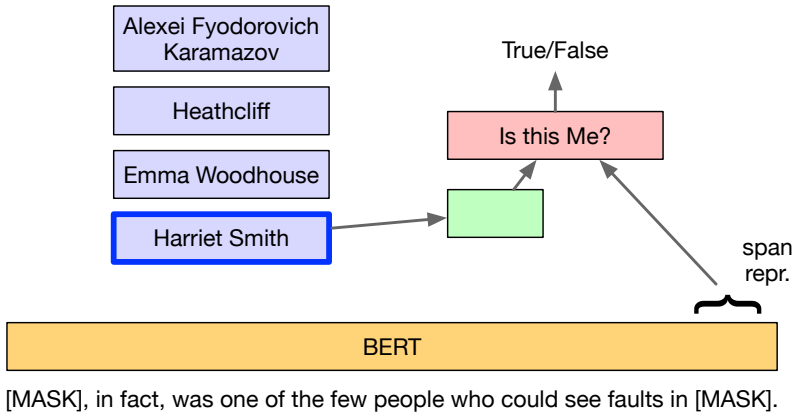

Figure 1: Sketch of the model. The sentence is from Jane Austen's "Emma". All characters are masked. In this case, the first character is Knightley, the second – the target – is Emma. Blue: entity library. Red: *Is this Me?* classifier.

there is enough information to train a representation.

As our intent is to induce entity representations for each character, we must mask each character mention. For each mention of any character predicted by BookNLP in each sentence in a novel, we replace the mention with a single [MASK] token in order to obscure the character's identity from the model. Multiword mentions are reduced to a single [MASK] token in order to prevent the model from being able to detect signal from mention length. Masking is applied to any mention, even for characters that appear fewer than 10 times.

For each sentence in a novel that contains at least one character mention, we produce at least two examples for the model: one positive example from the gold data, and one hallucinated example by randomly selecting a confound character from the same novel. If a character is mentioned more than one time in the same sentence, one mention is randomly selected to be the target mention for that character in that sentence. If a sentence talks about more than one character, a single positive example is generated for each character. Consider this sentence from Jane Austen's *Emma*:

> Whenever $[\text{MASK}]_{\text{James}}$ goes over to see $[\text{MASK}]_{\text{James'}}$ daughter, you know, $[\text{MASK}]_{\text{Miss Taylor}}$ will be hearing of us.

We first have to decide whether to first generate examples for James or for Miss Taylor. We pick one of the two at random, let us assume it is James. We next randomly select one of the two mentions of James to be the target mention. Let us say we pick the first. The input to the model for the *positive* example is then the Tenney et al. (2019) mix representation of the target mention concatenated with the current entity representation of James. We then construct a *negative* example by randomly selecting a character other than James to serve as a confound, following standard practice. If, for example, we were to sample Isabella (Emma's sister), the input to the model for the negative example from this mention would be the exact same mix representation of the target mention concatenated with the current entity embedding of the confound character, Isabella. With positive and negative examples constructed for James's mention, we then turn to the remaining character, Miss Taylor, and construct a positive and negative example for her mention.

Note that restricting the possible set of confounds for a given character to characters in the same novel, we have created a more difficult negative example than if we were to sample across all novels. For example, telling the difference between Elizabeth Bennet and Jane Bennet (both from Austen's *Pride and Prejudice*) is significantly more difficult than telling the difference between Elizabeth Bennet and the Cowardly Lion (from Baum's *The Wonderful Wizard of Oz*).

Training. All learned weights (the entity embeddings themselves, those in the projection layer, the scalars guiding the target mention representation, and those in the classifier) are updated with respect to cross entropy loss, optimized with Adam (Kingma and Ba, 2015) at a learning rate of 2e-05.

5 Experiments

5.1 Is this Me?

We first address our questions (a) and (b) from above: Is it possible to predict a masked mention of a literary character from an entity embedding, either within the same novel or in a summary of the same novel? And does performance degrade if we "skip ahead", using a character embedding trained on the beginning of a novel to predict a mention near the end?

5.1.1 Continuous Training

We begin by allowing the model to train entity representations (and all other learned weights) continuously throughout each novel. This means that we treat each example as a test example, and only allow the model to update based on its performance on a given example after its prediction has been made, as in a standard learning curve. As such, although the model is updated after every example, our performance statistics are computed over its prediction made *before* the update operation (meaning there is no performance computed over already-seen examples). As Table 2 shows, the model does well at this task, with overall accuracy across all characters and all novels of 74.37%. Accuracy was consistent across positive and negative examples. Most learning happened quickly within the first 50,000 examples, though accuracy did continue to increase through the entire run (Figure 2).

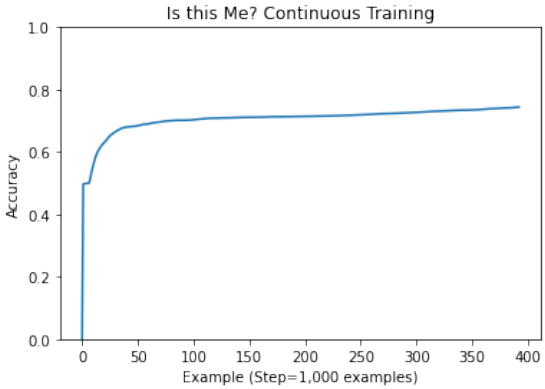

Figure 2: Is this Me? Continuous Training learning curve.

As should be expected, overall accuracy at the book level in this task is subject to frequency effects. Book-level accuracy exhibits strong positive correlation with the total number of examples per novel ($r = 0.584$; $p \ll 0.01$; Figure 3, left). Interestingly, however, book-level accuracy also increases with the number of characters modeled per book ($r = 0.500$; $p \ll 0.01$; Figure 3, right). To see whether the model is affected by language differences between older and more recent books, we used linear regression to predict book-level accuracy from novel publication date, finding very low correlation ($R^2 = 0.008$; $p = 0.663$).

	Examples	Correct	Accuracy
Positive	196,154	149,505	76.22%
Negative	196,154	142,258	72.52%
Total	392,308	291,763	74.37%

Table 2: *Is this Me?* accuracy for continuously trained entity representations.

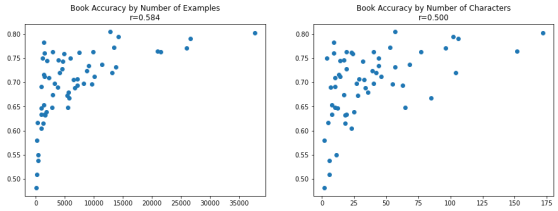

Figure 3: Is this Me? Continuous Training - Book-Level Accuracy. Accuracy within book (y-axis) is plotted against the number of examples for that book (x-axis).

At the character level, frequency effects were not nearly as strong, except in cases where characters were mentioned very frequently (defined here as characters with over 300 mentions). Across all characters, testing showed moderate positive correlation with mention frequency ($r = 0.174$; $p \ll 0.01$; Figure 4, left). Within frequently appearing characters, correlation with mention frequency was much higher ($r = 0.633$; $p \ll 0.01$; Figure 4, right).

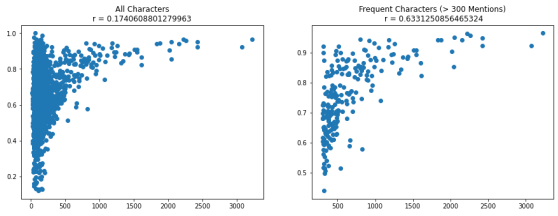

Figure 4: Is this Me? Continuous Training - Character-Level Accuracy. Accuracy within character (y-axis) is plotted against the number of examples for that character (x-axis).

5.1.2 Applicability to Novel Summaries

We also explored the extent to which the entity representations after having been trained on the full novel, could identify the same entities in short summaries of the same novel. To that end, we used the the WikiNovel summaries distributed with the Novel Aficionados dataset. The summaries show a strong domain shift compared to the novels. While they frequently do contain proper, if succinct, descriptions of the novel's plot and the involvement of major characters, they also exhibit significantly different patterns of language. Wikipedia entries do not just summarize the novels, they also frequently include metadiscursive language, as in this sentence from the WikiNovels summary of Austen's *Emma*:

> This point of view appears both as something perceived by *[emma_woodhouse]* an external perspective on events and characters that the reader encounters as and when *[emma_woodhouse]* recognises it and as an independent discourse appearing in the text alongside the discourse of the narrator and characters.

Because of this shift in domain, we see vastly reduced performance in character prediction and a heavy bias towards claiming the target mention is not a given character when using the model trained on the sentences from the novel. This is shown in Table 3. We evaluated the model in two settings. In the Pos. Only setting, all data points were positives, such that the model would have perfect accuracy by always saying yes. In the Pos. & Neg. setting, we use the same negative example generation technique as used in the model's training. While the model performs slightly better than chance when negative examples are included, it remains clear that future work should explore ways to generalize the entity representations such that they may be more informative across domain boundaries.

Example Types	Num. Examples	Accuracy
Pos. Only	7,736	36.12%
Pos. & Neg.	15,573	56.13%

Table 3: *Is this Me?* accuracy for continuously trained entity representations on WikiNovel summaries.

5.1.3 Non-Continuous Training

In §2 we noted that identifying masked character mentions is a not trivial due to the nature of narratives themselves. Literary plots are often constructed to force profound change in the behaviors, beliefs, and characteristics of central characters. This may be among the reasons that Bruera (2019) reported such difficulty with the Doppelgänger task. To see if distance in the novel affects our representations, we experimented with "skipping ahead" in the novel in order to determine the impact on performance when entities are not continuously updated.

Inspired by traditional character arcs, we split each novel into three sections of equal length (determined by number of sentences). The underlying assumption is that, due to the structure of the narrative, each character (especially main characters) will undergo some form of growth or change in between each novel section, suggesting that the learned entity representations should never be static in order to encode the results of that growth. We allowed a new *Is this Me?* classifier to learn representations for all literary entities using only the first third of the novels as training data, then froze the entity embeddings, and evaluated classifier performance against the middle and final thirds independently. We hypothesized that the model would exhibit a gradual decrease in performance as it moved further away from the point in time at which the entity representations were fixed, with the performance on the middle third better than performance toward the ends of the novels. Instead, we found a fairly rapid decline in performance (Table 4). Performance stays above chance, however, suggesting there is a kernel of each representation that is informative regardless. While this experiment does not explicitly demonstrate character development/change, the sharp decrease in performance when entity representations are fixed implicitly supports the claim that such change is present. Capturing that development directly, however, is another very difficult task and well-worthy of being the subject of future work.

Trained On	Beginning	Middle	End
Beginning	68.50%	55.70%	57.15%
Beg. & Mid.	68.50%	63.24%	57.45%

Table 4: *Is this Me?* accuracy on novels split into thirds.

5.2 Probing Entity Embeddings

We have found that, at least when entity embeddings are continuously trained, they can be used to predict a masked mention in a novel with reasonable accuracy. But are the resulting embeddings useful beyond the masked entity prediction task? To find this out, we turn to our questions (c) and (d) from above, and see if we can predict character gender from entity representation, and whether the identity of a character can be predicted from a description.

5.2.1 Predicting Gender from Literary Character Representation

We used a simple logistic regression model to probe the extent to which gender is encoded in the entity representations obtained from the continuous training in §5.1.1. As we have no gold annotation of literary character gender, we utilize the BookNLP preprocessing to look for gendered pronouns (*she/he*) for each character as a form of distant supervision. Manual inspection shows this heuristic to be very reliable after omitting characters for which no pronominal coreference link is available and characters who exhibit coreference chains featuring both gendered pronouns. This left a total of 2,195 characters (1,533 male, 662 female) to be considered for this experiment.

We learn a single weight for each embedding dimension for a total of 300 weights. In each case, we trained the classifiers on 80% of the characters across all novels (1,756 characters), leaving a test set of 439 characters. Each model was run four times, and we present the mean performance statistics in Table 5. Results were favorable across all runs, suggesting the learned character representations do encapsulate some knowledge of literary character gender.

μ **Acc**	μ **MSE**	μ **F1**
60.15%	0.3984	0.7208

Table 5: Model performance on predicting character gender from entity embeddings: Accuracy, mean squared error, and F1.

5.2.2 Character Descriptions

While the WikiNovels corpus is noisy and cluttered with metadiscursive literary commentary, as noted in §5.1.2, certain Wikipedia novel summaries do contain detailed descriptions of major characters. To better evaluate the ability of our learned entity representations to generalize outside of the domain of the novels on which they where trained, we manually extracted a subset of sentences which more readily pertained to our research question.

We isolated five novels which featured clean character descriptions within their summaries: Jane Austen's *Emma*, Charles Dickens's *A Tale of Two Cities* and *Great Expectations*, Fyodor Dostoevsky's *The Brothers Karamazov*, and Charlotte Brontë's *Jane Eyre*. From the character descriptions within these summaries we generated a total of 605 *Is this Me?*-style examples (positive and negative).[6] The pre-trained classifier exhibited performance above chance (61.63% accuracy), and a surprising ability to handle challenging out of domain sentences. While the model successfully predicted a high level description of Emma Woodhouse (Table 6; Row 1), it struggled with a similar description of Estella Havisham (Row 2). The model was also able to identify a character based on the description of a pivotal plot point (Row 3), but unsurprisingly struggled with more critical descriptions (Row 4).

6 Conclusion

In the ideal case, an entity embedding would constitute a compact representation of a person, their character traits and life story, and would allow for inferences about that person, including story arcs in which that person is likely to occur. What is the best way to learn embeddings for entities, and what can be learned from them? We have considered this question for the case of literary characters. We have trained entity embeddings through a masked prediction task, reading a collection of novels from begininng to end. We found that when trained continuously, the entity embeddings did well at the *Is this Me?* task: Given a target entity embedding, and given a sentence of the novel with a masked entity mention, is this a mention of the target entity? The *Is this Me?* task becomes much harder when we "skip ahead", training only on the first third of a novel and then evaluating on the middle and end. The task also becomes much harder when applied to Wikipedia summaries of novels, which show a marked domain difference from the novels themselves. Probing the entity embeddings that result from the masked prediction task, we find that they encode a good amount of information about the

[6] This set of examples may be found at http://www.katrinerk.com/home/software-and-data.

Novel	Target	Candidate	Result	Sentence
Emma	Emma	Emma	+	[MASK] the protagonist of the story is beautiful high spirited intelligent and lightly spoiled young woman from the landed gentry.
Great Expectations	Estella	Estella	-	She hates all men and plots to wreak twisted revenge by teaching [MASK] to torment and spurn men, including Pip who loves her.
A Tale of Two Cities	Miss Pross	Miss Pross	+	[MASK] permanently loses her hearing when the fatal pistol shot goes off during her climactic fight with Madame Defarge.
A Tale of Two Cities	Lucy Manette	Lucy Manette	-	She is the golden thread after whom book the second is named so called because [MASK] holds her father and her family lives together and because of her blond hair like her mother.

Table 6: Examples of the *Is this Me?* continuously trained classifier's performance on out-of-domain masked mentions found within the WikiNovels corpus. Non-target mentions have been de-masked for better readability.

entities. The gender of the literary character can in many cases be recovered from the embedding, and it is even often possible to identify a person from a Wikipedia description of their characteristic traits.

Looking ahead, the training regime and trained embeddings allow for many further analyses. We would like to probe further into the "skipping ahead" to better understand why it is so difficult. Intuitively, characters that undergo more development across the length of a novel should be more difficult to predict. It is not clear to what extent this is the case with the current model; this needs further study. In addition, we would like to model the change and development of characters more explicitly, for example by representing them as a trajectory over time rather than a single embedding. It would also be beneficial to further explore the ways in which character traits are implicitly present within entity representations learned from the *Is this Me?* task. While we attempted to probe this superficially via the evaluation on out-of-domain Wikipedia data, this data does not offer the annotation that would be necessary to perform a more in-depth analysis

We would also like to extend the model by including additional relevant input. At the moment, we essentially ask the model to bootstrap entity representations from scratch, using only the contextualized sentence representations produced by BERT and the current entity representations as input. Other useful information such as semantic relations (retrievable via dependency parse) may be useful. We may also consider the kind of events and modifiers that a given entity participates in to be able to exploit patterns across character archetypes (similar to Bamman et al. (2014)). We are also looking to extend the model to directly model relations between characters as relations between entity embeddings, to see whether this would help performance and to see to what extent the interpersonal relations of characters would be encoded in their embeddings.

Overall, we find the results presented in the current paper to be promising as a first step towards natural language understanding systems that use neural models of entities over time. As we have outlined here, however, there is still much work to be done.

7 Acknowledgements

This research was supported by the DARPA AIDA program under AFRL grant FA8750-18-2-0017. We acknowledge the Texas Advanced Computing Center for providing grid resources that contributed to these results, and results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation. We would like to thank the anonymous reviewers for their valuable feedback, as well as Jessy Li and Pengxiang Cheng.

References

David Bamman, Brendan OConnor, and Noah A Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 352–361.

David Bamman, Ted Underwood, and Noah Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference for Learning Representations (ICLR)*.

Andrea Bruera. 2019. Modelling the semantic memory of proper names with distributional semantics. Master's thesis, Universita degli Studi di Trento.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thibault Fevry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951. Association for Computational Linguistics.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1805–1816.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. 2017. Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1830–1839, Copenhagen, Denmark. Association for Computational Linguistics.

Oliver P John, Sanjay Srivastava, et al. 1999. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model.

Annie Louis and Charles Sutton. 2018. Deep Dungeons and Dragons: Learning character-action interactions from role-playing game transcripts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 708–713.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Isabel Briggs Myers and Peter B Myers. 2010. *Gifts Differing: Understanding Personality Type*. Nicholas Brealey.

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

Barbara Plank and Dirk Hovy. 2015. Personality traits on Twitter -or- how to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–98.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.