

Learning about human experiences through movie characters

Zhilin Wang¹ Weizhe Lin² Xiaodong Wu²

¹University of Washington, ²University of Cambridge
zhilinw@uw.edu, {wl356, xw338}@cam.ac.uk,

Abstract

While many different aspects of human experiences have been studied by the NLP community, none has captured its full richness. We propose a new task to capture this richness based on an unlikely setting: movie characters. We sought to capture theme-level similarities between movie characters that were community-curated into 20 thousand themes. By introducing a two-step approach that balances performance and efficiency, we managed to achieve 9-27% improvement over recent paragraph-embedding based methods. Finally, we demonstrate how the thematic information learnt from movie characters can be used to understand themes in the experience of people, as indicated on Reddit posts.

1 Introduction

What makes a person similar to another? While there is no definitive answer, some aspects that have been investigated in the NLP community are personality(Gjurković and Šnajder, 2018; Conway and O’Connor, 2016), demographics(Nguyen et al., 2016) as well as personal beliefs and intents (Sap et al., 2019). While each of these aspects is valuable on its own, they also seem somewhat lacking to sketch a complete picture of a person. Researchers who recognise such limitations seek to ameliorate them by jointly modelling multiple aspects at the same time (Benton et al., 2017). Yet, we intuitively know that as humans, we are more than the sum of the multiple aspects that constitutes our individuality. Our human experiences are marked by so many different aspects that interact in ways that we can not anticipate. What then can we do to better capture the degree of similarity between different people?

Finding similar movie characters can be an interesting first step to understanding humans better. Many characters are inspired by and related to true stories of people so understanding how to identify

similarities between character descriptions might ultimately help us to better understand similarities in human characteristics and experiences. One way of defining what makes movie character descriptions similar is when community-based contributors on All The Tropes¹ classify them into the same theme (also known as a trope), with an example from the trope “Driven by Envy” shown in Table 1. Other themes (tropes) include “Parental Neglect”, “Fallen Hero”, and “A Friend in Need”.

Such community-based curation allows All The Tropes to reap the same advantages as Wikipedia and open-sourced software: a large catalog can be created with high internal-consistency given the in-built self-correction mechanisms. This approach allowed us to collect a dataset of >100 thousand characters labelled with >20 thousand themes without requiring any annotation cost. Based on this dataset, we propose a model that can be used to identify similar movie characters precisely yet efficiently. While movie characters may not be the perfect reflection of human experience, we ultimately show that they are good enough proxies when collecting a dataset of similar scale with real people would be extremely expensive.

Superman’s 1990s enemy Conduit.

Conduit hates Superman because he knows if Superman wasn’t around he would be humanity’s greatest hero instead ...

Loki

Loki’s constant scheming against Thor in his efforts to one-up him gave Odin and the rest of Asgard more and more reasons to hate Loki ...

Table 1: Character descriptions from the trope “Driven by Envy”

¹<https://allthetropes.org>

Our key contributions are as follows:

1. We conduct a pioneering study on identifying similar movie character descriptions using weakly supervised learning, with potential implications on understanding similarities in human characteristics and experiences.
2. We propose a two-step generalizable approach that can be used to identify similar movie characters precisely yet efficiently and demonstrate that our approach performs at least 9-27% better than methods employing recent paragraph embedding-based approaches.
3. We show that our model, which is trained on identifying similar movie characters, can be related to themes in human experience found in Reddit posts.

2 Related Work

2.1 Analysis of characters in film and fiction

Characters in movies and novels have been computationally analyzed by many researchers. [Bamman et al. \(2013, 2014\)](#) attempted to cluster various characters into prototypes based on topic modelling techniques ([Blei et al., 2003](#)). On the other hand, [Fermann and Szarvas \(2017\)](#) and [Iyyer et al. \(2016\)](#) sought to cluster fictional characters alongside the relationships between them using recurrent neural networks and matrix factorization. While preceded by prior literature, our work is novel in framing character analysis as a supervised learning problem rather than an unsupervised learning problem.

Specifically, we formulate it as a similarity learning task between characters. Tapping on fan-curated movie-character labels (ie tropes) can provide valuable information concerning character similarity, which previous literature did not use. A perceptible effect of this change in task formulation is that our formulation allows movie characters to be finely distinguished amongst > 20000 themes versus < 200 in prior literature. Such differences in task formulation can contribute a fresh perspective into this research area and inspire subsequent research.

Furthermore, the corpus we use differs significantly from those used in existing research. We use highly concise character descriptions of around 200 words whereas existing research mostly uses movie/book-length character mentions. Concise

character descriptions can exemplify specific traits/experiences of characters. This allows the differences between characters to be more discriminative compared to a longer description, which might include more points of commonality (going to school/work, eating and having a polite conversation). This means that such concise descriptions can eventually prove more helpful in understanding characteristics and experiences of humans.

2.2 Congruence between themes in real-life experiences and movie tropes

Mostly researched in the field of psychology, real-life experiences are often analyzed through asking individuals to document and reflect upon their experiences. Trained analysts then seek to classify such writing into predefined categories.

[Demorest et al. \(1999\)](#) interpreted an individual's experience in the form of three key stages: an individual's wish, the response from the other and the response from the self in light of the response from the other. Each stage consists of around ten predefined categories such as wanting to be autonomous (Stage 1), being denied of that autonomy (Stage 2) and developing an enmity against the other (Stage 3). [Thorne and McLean \(2001\)](#) organized their analysis in terms of central themes. These central themes include experiences of interpersonal turmoil, having a sense of achievement and surviving a potentially life-threatening event/illness.

Both studies above code individuals' personal experiences into categories/themes that greatly resemble movie tropes. Because of this congruence, it is very likely that identifying similarity between characters in the same trope can inform about similarity between people in real-life. A common drawback of ([Demorest et al., 1999](#)) and ([Thorne and McLean, 2001](#)) lie in their relatively small sample size (less than 200 people classified into tens of themes/categories). Comparatively, our study uses $> 100,000$ characters fine-grainedly labelled by fans into $> 20,000$ tropes. As a result, this study has the potential of supporting a better understanding of tropes, which we have shown to be structurally similar to themes in real-life experiences.

2.3 Candidate selection in information retrieval

Many information retrieval pipelines involve first identifying likely candidates and then post-processing these candidates to determine which among them are most suitable. The most widely-

used class of approaches for this purpose is known as Shingling and Locally Sensitive Hashing (Leskovec et al., 2020; Rodier and Carter, 2020). Such approaches first represent documents as Bag-of-Ngrams before hashing such representation into shorter integer-vector signatures. These signatures contain information on n-gram overlap between documents and hence encode lexical features that characterize similar documents. However, such approaches are unable to identify documents that are similar based on abstract semantic features rather than superficial lexical similarities.

Recent progress in language modelling has enabled the semantic meaning of short paragraphs to be encoded beyond lexical features (Peters et al., 2018; Devlin et al., 2019; Howard and Ruder, 2018; Raffel et al., 2019). This has reaped substantial gains in text similarity tasks including entailment tasks (Bowman et al., 2015; Williams et al., 2018), duplicate questions tasks (Sharma et al., 2019; Nakov et al., 2017) and others (Cer et al., 2017; Dolan and Brockett, 2005). Yet, such progress has yet to enable better candidate selection based on semantic similarities. As a result, relatively naive approaches such as exhaustive pairwise comparisons and distance-based measures continue to be the dominant approach in identifying similar documents encoded into dense contextualized embeddings (Reimers and Gurevych, 2019). To improve this gap in knowledge, this study proposes and validates a candidate selection method that is compatible with recent progress in text representation.

3 Task formulation

There is a set of unique character descriptions from the All The Tropes ($Character_0, Character_1 \dots Character_n$), each associated with a non-unique trope (theme) ($Trope_0, Trope_1 \dots Trope_p$). Given this set, find the k (where $k = 1, 5$ or 10) most similar character(s) to each character without making explicit use of the trope association of each character. In doing so, the goal is to have a maximal proportion of most similar character(s) which share the same tropes.

4 Methods

In this section, we first discuss how we prepare the dataset and trained a BERT Next Sentence Prediction (NSP) model to identify similar characters. Based on this model, we present a 2-step **Select** and **Refine** approach, which can be utilized to find

the most similar characters quickly yet effectively.

4.1 Dataset

Character descriptions from All The Tropes² were used. We downloaded all character descriptions that had more than 100 words because character descriptions that are too short are unlikely to provide sufficient textual information for comparing similarity with other character descriptions. We then filtered our data to retain only tropes that contain more than one character descriptions. Character descriptions were then randomly split into training and evaluation sets (evaluation set = 20%). Inspired by BERT NSP dataset construction Devlin et al. (2019), we generated all possible combination-pairs of character descriptions that are classified under each trope and gave the text-pair a label of `IsSimilar`. For each `IsSimilar` pair in the training set, we took the first item, randomly selected a character description that is not in the same trope as the first item and gave the new pair a label of `NotSimilar`. This data is available at https://drive.google.com/file/d/1cF_NMP6yPGyGDQEwmkb8-cHKJB359Xka/view?usp=sharing

Descriptive statistics are available in Table 2.

4.2 Training BERT Next Sentence Prediction model

We trained a BERT Next Sentence Prediction model (English-base-uncased)³ with the pre-trained weights used as an initialization. As this model was trained to perform pair-wise character comparison instead of next sentence prediction, we thereafter name it as Character Comparison Model (CCM).

All hyper-parameters used to train the model were default⁴ except adjusting the maximum sequence length to 512 tokens (to adapt to the paragraph-length text), batch-size per GPU to 8 and epoch number to 2, as recommended by Devlin et al. (2019). Among the training set, 1% was separated as a validation set during the training process. We also used the default pre-trained BERT English-base-uncased tokenizer because only a small proportion of words ($< 0.5\%$) in the training corpus

²<https://allthetropes.org>

³12-layer, 768-hidden, 12-heads, 110M parameters with only Next Sentence Prediction loss, accessed from <https://github.com/huggingface/transformers>

⁴<https://github.com/huggingface/transformers/>

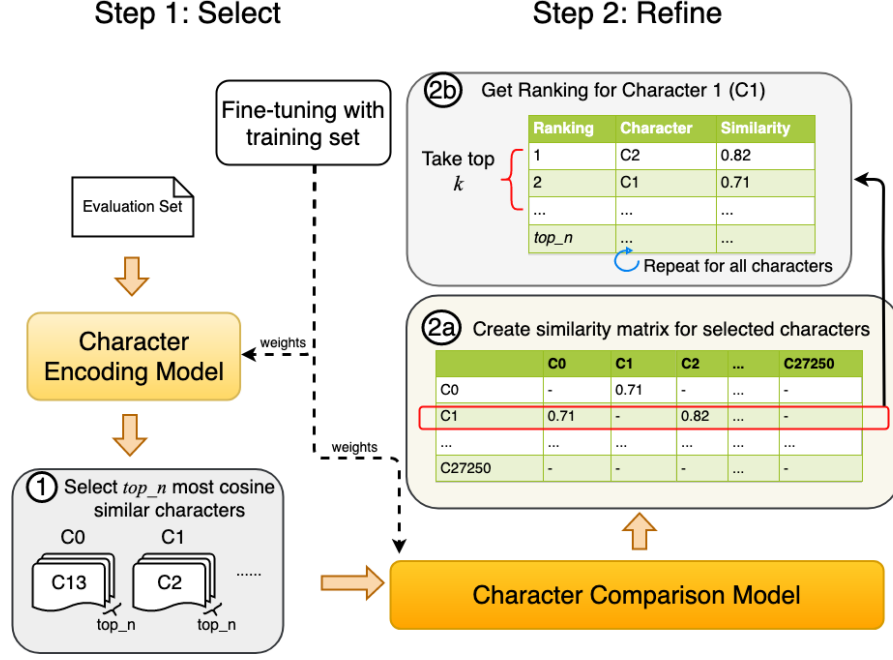


Figure 1: Workflow of finding most similar characters: BERT NSP model is first trained on the training set (Section 4.2) top_n characters are then selected using cosine similarity based on the Character Embedding Model or using a Siamese-BERT model, which has been omitted from the illustration for clarity (Section 4.3.1). This selection is then refined using the Character Comparison Model to create a similarity matrix, which can then be sorted to identified most similar characters. (Section 4.3.2)

	Training Set	Evaluation Set
Characters	109000	27250
Words per character	172 ($\sigma = 101$)	172 ($\sigma = 102$)
Tropes	13160	8669
Characters per trope	5.39 ($\sigma = 9.66$)	1.33 ($\sigma = 2.64$)
Character-pairs	2375298 (50% IsSimilar)	72705 (only IsSimilar)

Table 2: Descriptive statistics of dataset

were out-of-vocabulary, of which most were names. As a result, training took 3 days on 4 Nvidia Tesla P100 GPUs.

4.3 Select and Refine

To address the key limitation of utilizing exhaustive pairwise comparison in practice - its impractically long computation time (≈ 10 thousand GPU-hours on Nvidia Tesla P100), we propose a two-step Select and Refine approach. The Select step first identifies a small set of likely candidates in a coarse but computationally efficient manner. Then, the Refine step re-ranks these candidates using a precise but computationally expensive model. In doing so, it combines their strengths to precisely identify similar characters while being computationally efficient. While the Select and Refine approach is designed for identifying similar characters, this

novel approach can also be directly used in other tasks involving semantic similarities between a pair of texts.

4.3.1 Select

Characters that are likely to be similar to each character are first selected using a variant of our CCM model - named the Character Encoding Model (thereafter CEM). This model differs from the CCM model in that it does not utilize the final classifier layer. Therefore it can process a character description individually (instead of in pairs) to output an embedding that represents the character. The shared weights with CCM means that it encodes semantic information in a similar way. This makes it likely that the most cosine similar character descriptions based on their character embedding are likely to have high (but not necessarily the highest)

character-pair similarity.

Beyond the CEM, any model capable of efficiently generating candidates for similar character description texts in $O(n)$ time can also be used for this Select step, allowing immense flexibility in the application of the Select and Refine approach. To demonstrate this, we also test a Siamese-BERT model for the Select step, with the details of its preparation in Section 5.2.

In this step, we effectively reduced the search space for the most similar characters. We choose top_n candidates characters which are most similar to each character, forming top_n most similar character-pairs. top_n is a hyper-parameter that can range from 1 to 500. Strictly speaking, this step requires $O(n^2)$ comparisons to find the top_n most similar character-pairs. However, each cosine similarity calculation is significantly less computationally demanding compared to each BERT NSP operation (note that CCM is trained from an NSP model). This also applies to the Siamese-BERT model because character embeddings can be cached, meaning that only a single classification layer operation needs to be repeated $O(n^2)$ times. This means that computational runtime is dominated by $O(n)$ BERT NSP operations in the subsequent Refine step, given the huge constant factor for BERT NSP operations. Overall, this step took 0.25 GPU-hours.

4.3.2 Refine

The initial selection of candidates for most similar characters to each character will then be refined using the CCM model. This step is more computationally demanding ($0.25 * top_n$ GPU-hours) but can more effectively determine the extent to which characters are similar. Character Comparison Model (CCM) will then only be used on the top_n most similar candidate character-pairs, reducing the number of operations for each character from the total number of characters (n_{chars}) to only top_n . As a consequence, the runtime complexity of the overall operation is reduced from $O(n_{chars}^2)$ to $O(top_n \cdot n_{chars}) == O(n_{chars})$, given top_n is a constant.

5 Evaluation

In this section, we first present evaluation metrics and then present the preparation of baseline models including state-of-the-art paragraph-level embedding models. Finally, we analyze the performance of our models relative to baseline models.

5.1 Evaluation metrics

Recall @ k considers the proportion of all ground-truth pairs found within the k (1, 5 or 10) most similar characters to each character (Manning et al., 2008). Normalized Discounted Cumulative Gain @ k (**nDCG @ k**) is a precision metric that considers the proportion of predicted k most similar characters to each character that are in the ground-truth character-pairs. It also takes into account the order amongst top k predicted most similar characters (Wang et al., 2013). Mean reciprocal rank (**MRR**) identifies the rank of the first correctly predicted most similar character for each character and averages the reciprocal of their ranks. (Voorhees, 2000). Higher is better for all metrics.

5.2 Baseline Models

Baseline measurements were obtained for Google Universal Sentence Encoder-large (Cer et al., 2018), BERT-base (Devlin et al., 2019) and Siamese-BERT-base⁵ (Reimers and Gurevych, 2019).

Google Universal Sentence Encoder-large model⁶ (**USE**) on Tensorflow Hub was used to obtain a 512-dimensional vector representation of each character description. Bag of Words (**BoW**) was implemented by lowercasing all words and counting the number of times each word occurred in each character description. **BERT** embedding of 768 dimensions were obtained by average-pooling all the word embedding of tokens in the second-to-last layer, as recommended by (Xiao, 2018). The English-base-uncased version⁷ was used. For each type of embedding, the most similar characters were obtained by finding other characters whose embeddings are most cosine similar.

Siamese-BERT was obtained based on training a Siamese model architecture connected to a BERT base model on the training set in Section 4.1. We follow the optimal model configuration for sentence-pair classification tasks described in Reimers and Gurevych (2019), which involves taking the mean of all tokens embeddings in the final layer. With the mean embedding for each character description, an absolute difference between them was taken. The mean embedding for character A, mean embedding for character B and their

⁵ 12-layer, 768-hidden, 12-heads and 110M parameters

⁶ <https://tfhub.dev/google/universal-sentence-encoder-large/3>

⁷ 12-layer, 768-hidden, 12-heads and 110M parameters

absolute difference was then entered into a feedforward neural network, which makes the prediction. Siamese-BERT was chosen as a baseline due to its outstanding performance in sentence-pair classification tasks such as Semantic Textual Similarity (Cer et al., 2017) and Natural Language Inference (Bowman et al., 2015; Williams et al., 2018). For this baseline, the characters most similar to a character are those with the highest likelihood of being predicted `IsSimilar` with the character.

5.3 Suitability of Siamese-BERT and CEM for Step 1: Select

While the prohibitively high computational demands of exhaustive pairwise comparison (≈ 10 thousand GPU-hours) prevents a full-scale evaluation of the adequateness of Siamese-BERT and CEM for Step 1:Select, we conducted a small-scale experiment on 100 randomly chosen characters from the test set. First, an exhaustive pairwise comparison was conducted between these randomly chosen characters and all characters in the test set. From this, 100 characters with the highest CCM similarity value with each of the randomly chosen characters were identified. Next, various methods in Table 3 were attempted to identify 500 characters with the highest cosine similarity with the randomly chosen characters. Finally, the proportion of overlap between CCM and each method was calculated. Results demonstrate that Siamese-BERT and CEM have the greatest overlap and hence, the use of Siamese-BERT and CEM can select for the most number of highly similar characters to be refined by the CCM.

	CCM overlap (%)
Siamese-BERT	36.15
CEM	24.21
BERT	16.90
USE	16.27
BoW	7.41

Table 3: Proportion of 100 characters with high CCM similarity value that overlaps with each method for Step 1: Select

5.4 Selecting hyper-parameter top_n for Step 2: Refine

Based on Figure 2, the ideal top_n for the Select and Refine model with Siamese-BERT varies between 7 and 25 depending on the metric that is optimised for. In general, a lower value for top_n

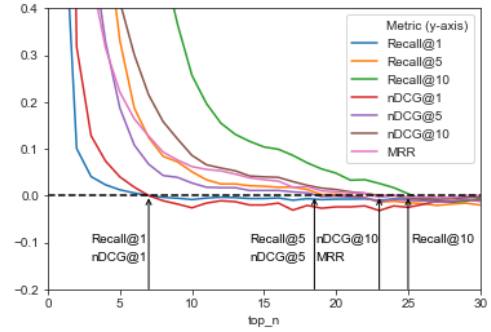


Figure 2: Percent change in metrics with each additional top_n for Select and Refine model with Siamese-BERT. Average smoothing applied over a range of 10 to improve clarity. Points annotated where each metric is at 0.

is preferred when optimizing for Recall@k and nDCG@k with smaller values of k. The metrics reported in Table 4 consist of the optimal value for each metric at various top_n .

On the other hand, there is no ideal value for top_n when using the Select and Refine model with CEM. Instead, the metrics continue to improve over large values of top_n , albeit at a gradually reduced rate. However, due to practical considerations relating to GPU computation time, we terminated our search at $top_n = 500$ and report metrics for that value of top_n .

Together, this means that the Select and Refine model using Siamese-BERT achieves peak performance with significant less computational resources compared to the one using CEM (2-6 GPU-hours vs. 125 GPU-hours).

5.5 Comparing Select and Refine models with baseline models

As shown in Table 4, the highest value for all metrics lies below 40% suggesting that identifying similar characters is a novel and challenging task. This is because there are only very few correct answers (characters from the same trope) out of 27,000 possible characters. The poor performance of the Bag-of-Words baseline also demonstrates that abstract semantic similarity between characters is significantly different from their superficial lexical similarity. In face of such challenges, the Select and Refine model using Siamese-BERT performed 9-27 % better on all metrics than the best performing paragraph-embedding-based baseline. This suggests the importance of refining initial selection of candidates instead of using them directly,

	Recall @ k (in %)			nDCG @ k (in %)			MRR (in %)
	k = 1	k = 5	k = 10	k = 1	k = 5	k = 10	
Select and Refine models							
Siamese-BERT	6.921	23.53	36.14	21.82	19.13	19.71	26.56
CEM	6.184	20.74	31.02	19.50	17.02	17.26	28.50
Baseline models							
Siamese-BERT	5.437	19.53	30.65	17.14	15.51	16.15	25.99
CEM	2.802	8.852	13.26	8.832	7.119	7.126	14.61
BERT	1.238	3.636	5.514	3.904	3.035	3.109	7.182
USE	1.277	4.427	6.956	4.025	3.599	3.866	7.810
BoW	0.4632	1.344	1.987	1.46	1.087	1.052	2.824

Table 4: Performance of Select and Refine models compared to baseline models. Higher is better for all metrics.

even when the baseline model has relatively good performance.

Comparing the Select and Refine models, Siamese-BERT performed much better than CEM while having a significantly low *top_n*, which means that less computational resources is required. The superior performance and efficiency of Siamese-BERT means that it is more suitable for Step 1: Select. This is likely caused by the higher performance of Siamese-BERT as a baseline model. While it was surprising that using Siamese-BERT outperformed CEM, which directly shares weights with the CCM, such an observation also shows the relatively low coupling between the Select and Refine steps. This means that the Select and Refine approach that we propose can continue to be relevant when model architectures that are more optimized for each step are introduced in the future.

The significantly higher performance of Select and Refine models can be attributed to the ability of underlying BERT NSP architecture in our CCM to consider complex word relationships across the two character descriptions. A manual examination of correct pairs captured only by Select and Refine models but not baseline models revealed that these pairs often contain words relating to multiple common aspects. As an example, one character description contains “magic, enchanter” and “training, candidate, learn” while the other character in the ground-truth pair contains “spell, wonder, sphere” and “researched, school”. Compressing these word-level aspects into a fixed-length vector would cause some important semantic information - such as the inter-relatedness between aspects - to be lost (Conneau et al., 2018). As a result, cap-

turing similarities between these pairs prove to be difficult in baseline models, leading to sub-optimal ranking of the most similar characters.

6 Implications for understanding themes in real-life experiences

6.1 Relating movie characters to Reddit posts

To demonstrate the potential applications of this study in understanding human experiences, we designed a task that can show how the model can be used with zero-shot transfer learning. Specifically, we used our model to identify the movie-characters that are most fitting to a description of people’s life experiences. To do this, we collected 50 posts describing people’s real-life experiences from a forum r/OffMyChest on Reddit⁸, on which people share their life experiences with strangers online.

Then, we used models to identify 10 movie characters (from our test set) that are most befitting to each post. For each of these 10 movie characters suggested by model, three of the authors independently rated whether the character matches the concepts, ideas and themes expressed in each post, while blind to information on which model the characters were generated by. Because the extent of similarity between a movie character and a Reddit post can be ambiguous, a binary annotation was chosen over a Likert scale for clarity of annotation. Annotators were instructed to annotate “similar” when they can **specify** at least one area of overlap between the concepts, ideas and themes of a Reddit post and a movie character. Examples of some characters that are indicated as “similar”

⁸<https://www.reddit.com/r/offmychest/>

	Precision @ k (in %)		
	k = 1	k = 5	k = 10
Select and Refine models			
Siamese-BERT	98.00 (14.00)	92.40 (14.42)	87.00 (8.797)
CEM	82.00 (39.56)	77.60 (17.87)	70.20 (8.944)
Baseline models			
Siamese-BERT	76.00 (42.78)	73.20 (14.94)	70.80 (8.306)
CEM	48.00 (38.41)	33.20 (20.48)	27.20 (11.34)
BERT	40.00 (48.99)	21.20 (12.65)	12.80 (5.978)
USE	32.00 (46.65)	15.60 (13.26)	9.20 (7.231)
BoW	16.00 (36.66)	7.20 (9.166)	4.40 (4.898)

Table 5: Precision @ k (std. dev.) for movie characters identified by each model.

to two posts are shown in Appendix A. All annotators agree on 94.2% of labels, which is very high. Where the annotators disagree, the majority opinion out of three is taken. From these annotations, **Precision @ k** is calculated, considering the proportion of all characters identified within the k (1, 5 or 10) that are labelled as "similar" (Manning et al., 2008).

In Table 5, the performance of our Select and Refine models reflects a similar extent of improvement compared to our main learning task. This shows that the model that was trained to disambiguate movie character similarity can also determine the extent of similarity between movie characters and people’s life experiences. Beyond the relative performance gains, the Select and Refine model on this task also demonstrates an excellent absolute performance of precision @ 1 = 98.00 %. This means that it can be used on this task and potentially many other related tasks without any fine-tuning.

6.2 Future directions

We are excited about the diversity of research directions that this study can complement. One possible area is social media analysis (Zirikly et al., 2019; Amir et al., 2019; Hauser et al., 2019). Researchers can make use of movie characters with known experiences (e.g. mental health, personal circumstances) to identify similar experiences in social media when collecting large amounts of text labelled with such experiences directly is difficult.

Another area would be personalizing dialogue agents (Tigunova et al., 2020; Zhang et al., 2018). In the context of limited personality-related train-

ing data, movie characters with personality that are similar to a desired dialogue agent can be found. Using this, a dialogue agent can be trained with movie subtitle language data (involving the identified movie character). Thereby, the augmented linguistic data enables the dialogue agent to have a well-defined, distinct and consistent personality.

A final area that can benefit from this study is media recommendations (Rafailidis et al., 2017). Users might be suggested media content based on the extent to which movie characters resonate with their own/friends’ experiences. Additionally, with social environments being formed in virtual reality (Chu et al., 2020), participants can even assume the identity of movie characters that they are similar to, so as to have an interesting and immersive experience.

7 Conclusion

We introduce a pioneering study on identifying similar movie characters through weakly supervised learning. Based on this task, we introduce a novel Select-and-Refine approach that allows us to match characters belonging to a common theme, which simultaneously optimize for efficiency and performance. Using this trained model, we demonstrate the potential applications of this study in identifying movie characters that are similar to human experiences as presented in Reddit posts, without any fine-tuning. This represents an early step into understanding the complexity and richness of our human experience, which is not only interesting in itself but can also complement research in social media analysis, personalizing dialogue agents and media recommendations/interactions.

References

- Silvio Amir, Mark Dredze, and John W. Ayers. 2019. [Mental health surveillance over social media with digital cohorts](#). In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 114–120, Minneapolis, Minnesota. Association for Computational Linguistics.
- David Bamman, Brendan O'Connor, 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)*, pages 352–361.
- David Bamman, Ted Underwood, and Noah A 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)*, pages 370–379.
- Adrian Benton, Margaret Mitchell, and Dirk Hovy. 2017. [Multitask learning for mental health conditions with limited social media data](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 152–162, Valencia, Spain. Association for Computational Linguistics.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. [Latent dirichlet allocation](#). *J. Mach. Learn. Res.*, 3:993–1022.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.
- Hang Chu, Shugao Ma, Fernando De la Torre, Sanja Fidler, and Yaser Sheikh. 2020. Expressive telepresence via modular codec avatars. In *Computer Vision – ECCV 2020*, pages 330–345, Cham. Springer International Publishing.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single \\$&!#* vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- Mike Conway and Daniel O'Connor. 2016. [Social media, big data, and mental health: current advances and ethical implications](#). *Current Opinion in Psychology*, 9:77 – 82. Social media and applications to health behavior.
- Amy Demorest, Paul Crits-Christoph, Mary Hatch, and Lester Luborsky. 1999. A comparison of interpersonal scripts in clinically depressed versus nondepressed individuals. *Journal of Research in Personality*, 33(3):265–280.
- 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.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Lea Frermann and György Szarvas. 2017. Inducing semantic micro-clusters from deep multi-view representations of novels. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1873–1883.
- Matej Gjurković and Jan Šnajder. 2018. [Reddit: A gold mine for personality prediction](#). In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 87–97, New Orleans, Louisiana, USA. Association for Computational Linguistics.
- Michael Hauser, Evangelos Sariyanidi, Birkan Tunc, Casey Zampella, Edward Brodtkin, Robert Schultz, and Julia Parish-Morris. 2019. [Using natural conversations to classify autism with limited data: Age matters](#). In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 45–54, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544.

- Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2020. *Finding Similar Items*, 3 edition, page 78–137. Cambridge University Press.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.
- Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. *SemEval-2017 task 3: Community question answering*. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada. Association for Computational Linguistics.
- Dong Nguyen, A. Seza Doğruöz, Carolyn P. Rosé, and Franciska de Jong. 2016. *Computational sociolinguistics: A survey*. *Computational Linguistics*, 42(3):537–593.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- D. Rafailidis, P. Kefalas, and Y. Manolopoulos. 2017. *Preference dynamics with multimodal user-item interactions in social media recommendation*. *Expert Systems with Applications*, 74:11 – 18.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.
- Nils Reimers and Iryna Gurevych. 2019. *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Simon Rodier and Dave Carter. 2020. *Online near-duplicate detection of news articles*. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1242–1249, Marseille, France. European Language Resources Association.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. *Atomic: An atlas of machine commonsense for if-then reasoning*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3027–3035.
- Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. *Natural language understanding with the quora question pairs dataset*. *CoRR*, abs/1907.01041.
- Avril Thorne and Kate C. McLean. 2001. *Manual for Coding Events in Self-Defining Memories*. Unpublished Manuscript.
- Anna Tigunova, Andrew Yates, Paramita Mirza, and Gerhard Weikum. 2020. *CHARM: Inferring personal attributes from conversations*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5391–5404, Online. Association for Computational Linguistics.
- Ellen M Voorhees. 2000. The trec-8 question answering track report. Technical report.
- Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *COLT*.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. *A broad-coverage challenge corpus for sentence understanding through inference*. 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 1112–1122. Association for Computational Linguistics.
- Han Xiao. 2018. bert-as-service. <https://github.com/hanxiao/bert-as-service>.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. *Personalizing dialogue agents: I have a dog, do you have pets too?* In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. *CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts*. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 24–33, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix A

Reddit post	My father passed away when I was 6 so I didn't really remember much of him but the fact that I didn't recognize his picture saddens me.
Movie characters	<p>1. Sisko in Star Trek: Deep Space Nine (Past Tense) When he encountered an entry about the historical figure, passed comment about how closely Sisko resembled a picture of him (the picture, of course, being that of Sisko.)</p> <p>2. Arator the Redeemer in World of Warcraft As Arator never knew his father, he asks several of the veteran members of Alliance Expedition about Turalyon for information and leads on Turalyon's current location. Several people then gave their opinion on how great a guy Turalyon was, but sadly, he has been MIA for 15 years.</p> <p>3. Kira in Push The reality of a photo taken at Coney Island is the key evidence that causes her to realize that this was a fake memory.</p> <p>4. Todd Aldridge in Mindwarp Todd shows up back in town; to him, there was a bright light one night, and he returned several months later with no knowledge of the intervening period.</p> <p>5. Parker Girls in Stranger in Paradise However, when the operation collapsed after the death of Darcy Parker many Parker Girls were trapped in their cover identities, unable to extricate themselves from the lives they had established.</p>
Reddit post	The black ladies I work with make me feel the most loved I've felt in years. I've had a horrible past 10 years. Childhood trauma and depression, addiction, abuse etc
Movie characters	<p>1. Shinjiro Aragaki in Persona 3 First of all, he's an orphan. During those two years, he began taking drugs to help control his Persona. Said drugs are slowly killing him. He has his own Social Link with the female protagonist where it becomes painfully clear that he really is a nice guy, and he slowly falls in love with her.</p> <p>2. Mami in Breath of Fire IV Country Mouse finds King in the Mountain God-Emperor that The Empire (that aforementioned God-Emperor founded) is trying very, very hard to kill. Country Mouse Mami nurses God-Emperor Fou-lu back to health. Mami and Fou-lu end up falling in love.</p> <p>3. Emi in Katawa Shoujo The loss of her legs was traumatic, but she learned to cope with that well. The loss of her dad she did not cope with at all. Part of getting her happy ending is to help her deal with her loss.</p> <p>4. Harry in Harry Potter Harry reaches out, has friends, and even in the moments when the school turns against him, he still has a full blown group of True Companions to help him, thus making him well adjusted and pretty close to normal.</p> <p>5. Commander Shepard in the Mass Effect series If the right dialogue is chosen, s/he's cynical and bitter with major emotional scars from his/her past experiences. It becomes pretty clear how emotionally burned out s/he really is.</p>

Table 6: Excerpts from Posts from Reddit r/OffMyChest to five similar movie characters. Excerpts of Reddit posts mildly paraphrased to protect anonymity.