DOCENT: Learning Self-Supervised Entity Representations from Large Document Collections

Yury Zemlyanskiy[*]
USC
zemlyans@usc.edu

Sudeep Gandhe
Google Research
srgandhe@google.com

Ruining He
Google Research
ruininghe@google.com

Bhargav Kanagal
Google Research
bhargav@google.com

Anirudh Ravula
Google Research
braineater@google.com

Juraj Gottweis
Google Research
juro@google.com

Fei Sha[†]
Google Research
fsha@google.com

Ilya Eckstein
Google Research
ilyaeck@google.com

Abstract

This paper explores learning rich self-supervised entity representations from large amounts of associated text. Once pre-trained, these models become applicable to multiple entity-centric tasks such as ranked retrieval, knowledge base completion, question answering, and more. Unlike other methods that harvest self-supervision signals based merely on a local context within a sentence, we radically expand the notion of context to include *any* available text related to an entity. This enables a new class of powerful, high-capacity representations that can ultimately distill much of the useful information about an entity from multiple text sources, without any human supervision.

We present several training strategies that, unlike prior approaches, learn to *jointly* predict words and entities—strategies we compare experimentally on downstream tasks in the TV-Movies domain, such as MovieLens tag prediction from user reviews and natural language movie search. As evidenced by results, our models match or outperform competitive baselines, sometimes with little or no fine-tuning, and can scale to very large corpora.

Finally, we make our datasets and pre-trained models publicly available[1]. This includes *Reviews2Movielens*, mapping the ~1B word corpus of Amazon movie reviews (He and McAuley, 2016) to MovieLens tags (Harper and Konstan, 2016), as well as Reddit Movie Suggestions with natural language queries and corresponding community recommendations.

Review 1: *"This movie develops its power best if you don't try to look out for the "real" and "true" events behind the four versions of the narration... shown in a very intelligent and artistic way, no silly plot-twists, no explanation in the end — it is open to your fantasy... "*$MOVIE*" is an important piece of cinematic storytelling and a really interesting way to reflect on the origin of tales... Some scenes even remind me of Andrej Tarkovskijs intensive style..".*
Review 2: *"Just rented this, and at first I didn't like very much, but then it starts to sink in for how good it is, the acting is great especially Toshiro Mifune, it was shot very good for an older movie... it's #62 on the top 250"*
Review 3: *"Saw this movie at my local video store... was placed on a waiting list, but when I returned to check it out the video store had closed down over night. Actually whent out of business"*
... More reviews ...
Summary tags: [nonlinear] [multiple storylines] [japan] [black and white] [surreal] [cerebral] [imdb top 250], ...

Table 1: *Reviews2Movielens* task, illustrated. Here are sample review snippets for a certain classic film which is summarized using MovieLens tags. Notice that the tags may not appear in the input verbatim and can be thought of as boolean questions about the film. Note also that Review 3 has zero relevant signal—a common challenge of low SNR in this dataset. Bonus teaser: can you guess the $MOVIE from these snippets? This little quiz alludes to a key learning task in our approach.

1 Introduction

Much of the online information describing entities in domains such as music, movies, venues or consumer products, is only available as unstructured text—a format that is human-readable but not machine-understandable (yet). Consider online reviews—a rich source of mostly user-generated about a vast number of entities. Our key research question is: *Can we learn strong models for entity understanding tasks such as vertical search and question answering, solely from text?* In other words, given a large and noisy collection of documents about an entity, can we distill all the useful information therein into a dense entity representa-

[*] Work is partially done while at Google

[†] On leave from USC (feisha@usc.edu)

[1] See http://goo.gle/research-docent for *Reviews2Movielens* and models. Scripts and *Reddit Suggestions* can be found at https://urikz.github.io/docent

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2540–2549
April 19 - 23, 2021.

tion, so as to benefit multiple downstream tasks?

Traditionally, learning entity representations required supervised signals such as clicks, "likes" and consumption behavior (Agichtein et al., 2006; Huang et al., 2013; Koren et al., 2009; Vig et al., 2012a), which are generally expensive and time consuming to obtain at scale. To leapfrog these limitations, we draw inspiration from the recent progress in unsupervised learning of text, particularly contextualized representations via techniques such as ELMo (Peters et al., 2018), CoVe (McCann et al., 2017) and BERT (Devlin et al., 2019). Many of these representations are learned by predicting a missing word from its context. More recently, Sun et al. (2019) showed that extending word masking strategies to entities can lead to superior language models. Even more recent entity linking methods such as RELIC (Ling et al., 2020) and others, detailed in Section 6, were shown to produce explicit encodings applicable to entity understanding tasks.

We start with RELIC-like approaches and generalize them into a family of models, collectively called DOCENT, that jointly embed text and entities (Section 2) via self-supervised tasks. The first one, DOCENT-DUAL, is essentially RELIC, but trained with a much broader context to include any and all sentences potentially related to an entity. Importantly, DOCENT-DUAL/RELIC only optimizes a single task, namely *entity prediction* given an associated sentence, effectively modeling $P(Entity|Sentence)$.

Another natural way of jointly modelling entities and text is by directly tapping the cross-attention mechanism in BERT, simply by extending the BERT vocabulary to include entity tokens V_E. Each entity-related sentence can then be augmented with a corresponding token from V_E. We call this method DOCENT-FULL and, despite (or perhaps because of) its conceptual simplicity, it proves surprisingly effective in semi-supervised tasks.

Finally, DOCENT-HYBRID aims to capture the best of both models by extending DOCENT-DUAL with an additional task of predicting words in a sentence, conditioned on its associated entity. This task encourages the latter to "remember" salient phrases in its sentences.

We empirically evaluate these methods by learning entity representations for movies from a TV-Movies portion of the Amazon Reviews Corpus (He and McAuley, 2016). To this end, we consider several movie-oriented tasks for downstream evaluation, i.e. Reddit Movie Suggestions and MovieLens Tag Prediction (Harper and Konstan, 2016), which we study in both zero-shot, supervised and few-shot settings. We join the MovieLens dataset with the reviews corpus (He and McAuley, 2016) obtaining a mapping from movie reviews to user-generated tags. On the supervised tag prediction task, our text-based model demonstrates SOTA performance, despite not using powerful user signals (Vig et al., 2012a). In fact, we are able to match or outperform baselines on all tasks where they are available.

1.1 Contributions

1. First, we propose a family of methods to train deep self-supervised entity representations purely from related text documents, with strong zero-shot results on ranked retrieval with natural language queries.

2. Secondly, we show that these pre-trained representations are amenable to fine-tuning on new tasks such as MovieLens tag prediction, where we show state-of-the-art results. They are also effective few-shot learners, which we demonstrate on a harder *open-vocabulary*[2] task akin to Boolean Question Answering (Clark et al., 2019).

3. Next, we propose *Reviews2Movielens*—a new Text Based Entity Understanding task. The requisite dataset, which we release publicly, effectively joins the Amazon Movie Reviews Corpus and MovieLens into a large, sparsely supervised set with approximately 1B words and 470K movie-tag pairs.

4. Finally, we also release a dataset of user-generated Reddit Movie Suggestions, a benchmark for natural language search and recommendation scenarios.

2 Self-Supervised Entity Representations

Inspired by the success of self-supervised language models, we seek to extend them to jointly compute text and entity representations. Recall that our input is a set of entities $\mathcal{E}$ where for every entity $e \in \mathcal{E}$, we have a collection of sentences, denoted by $\mathcal{S}_e$, from all documents related to e. Intuitively, we want the representation of e to be influenced by each associated sentence $s \in \mathcal{S}_e$, and vice versa.

[2] An open vocabulary allows any phrase to be a label.

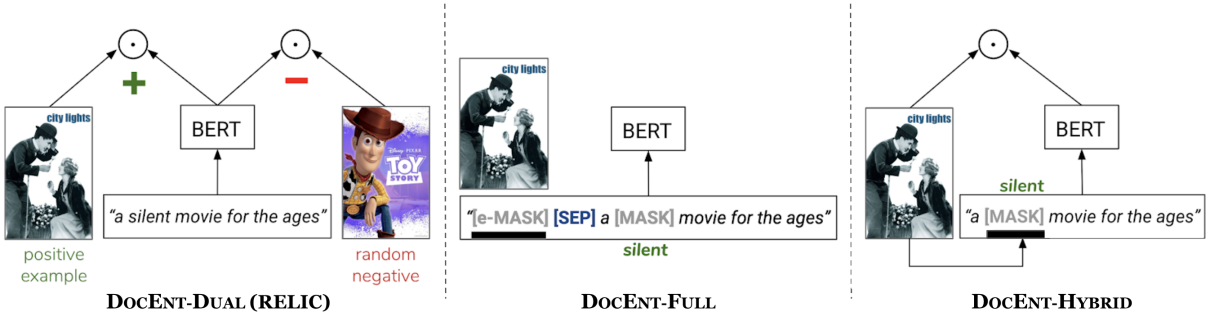


Figure 1: Models in the DocEnt family. Left: a baseline dual encoder model called DocEnt-Dual a.k.a. RELIC, maximizing $P(e|s)$ but not $P(s|e)$. Center: DocEnt-Full—a model maximizing the joint sentence-entity probability using full cross-attention. Right: DocEnt-Hybrid, designed to capture the best of both worlds.

To that end, we explore two (self-) supervision signals: $P(e \mid s)$ and $P(s \mid e)$.

2.1 DocEnt-Dual, Known as RELIC

At the core of DocEnt-Dual is a RELIC model that co-encodes an entity e and an associated sentence $s \in \mathcal{S}_e$ so as to maximize their compatibility score, defined as the cosine similarity between the two encodings:

$$\mathbf{s}(e, s) = \frac{g(e)^T f_{CLS}(s)}{\|g(e)\| \|f_{CLS}(s)\|},$$

where $g(e)$ is an embedding of e and $f(s)$ is a BERT-based encoding of s, with its special $[CLS]$ token whose output representation is denoted by f_{CLS}. Then, the conditional probability of e given s is given by a softmax over the set $\mathcal{E}$ [3]:

$$P(e|s) = \frac{\exp(\mathbf{s}(e, s))}{\sum_{e' \in \mathcal{E}} \exp(\mathbf{s}(e', s))}.$$

Finally, RELIC is trained by maximizing $\log P(e|s)$ over all associated pairs $e, s \in \mathcal{S}_e$:

$$\mathcal{L}_E(e, s) = \log P(e|s).$$

Note that both g and f (initialized with a common BERT) are learned during training.

Our sole difference to the original RELIC is in training data: while RELIC only uses sentences containing entity mentions, we allow a radically broader context – all sentences associated with an entity – with the goal of remembering all of its attributes. Crucially, no human labeling is required.

Despite its effectiveness (as demonstrated in Section 5), RELIC has one obvious limitation: it ignores $P(s \mid e)$, leaving a useful signal "on the table". We therefore propose another way of co-encoding sentences and entities by tapping the full cross-attention power of Transformers.

2.2 DocEnt-Full

Before we proceed, let us revisit BERT's Masked Language Model (MLM) training objective. Given a sequence of input tokens $s = [s_1, \ldots, s_n]$, a fraction of tokens s_J at randomly selected positions J is replaced with a special *[MASK]* token. We denote this new sequence by s_{-J}.

Then, BERT predicts masked tokens based on their contextualized representations $f(s_{-J})$. The MLM training objective to maximize is:

$$\mathcal{L}_{MLM} = \log P(s_J \mid s_{-J}).$$

Enter DocEnt-Full. It follows the standard BERT architecture, with a twist. First, we expand the input vocabulary to include all entity tokens in $\mathcal{E}$. Then, during input sequence construction, each sentence $s \in \mathcal{S}_e$ is prepended[4] with the corresponding entity token e, as shown in Figure 1. This way, masking and predicting this token (via softmax) effectively adds our new objective $\mathcal{L}_E$ to BERT. Further, the new e token is now part of a sentence context, augmenting the original $\mathcal{L}_{MLM}$ to

$$\mathcal{L}_{MLM+E}(s, e) = \log P(s_J \mid s_{-J}, e),$$
$$\text{and } \mathcal{L}_{\text{Full}} = \mathcal{L}_E + \lambda \mathcal{L}_{MLM+E}$$

becomes the combined loss function optimized using nothing but BERT's standard MLM training, with a hyperparameter λ to balance the two terms[5].

[3] In practice, only a subset of entities in $\mathcal{E}$ is used in the denominator: the so called "in-batch negatives".

[4] Technically, we replace BERT's standard (s_A, s_B) two-segment input structure with (e, s), for $s \in \mathcal{S}_e$.

[5] The relative masking frequency of entity tokens is another hyperparameter available to balance the two objectives.

This conceptual simplicity and full cross-attention power come with a cost: bundling word-pieces and entities together forces the model to allocate an equal capacity to both types of tokens (e.g., 768D for BERT-base), regardless of the size of $\mathcal{E}$. As a result, a relatively small-sized $\mathcal{E}$ may be prone to overfitting[6] in zero-shot scenarios, as we observe in Section 5.4.2.

2.3 DocEnt-Hybrid

Recall that RELIC avoids the above limitation by decoupling text and entity encoders. To get the best of both worlds, we introduce DocEnt-Hybrid—a third model that sticks with the modular dual encoder architecture while also modeling $P(s \mid e)$. This is achieved by implementing a different variant of $\mathcal{L}_{MLM+E}$ where, for every masked wordpiece token, the output of Transformer layers $f(s_{-J})$ is first concatenated with the associated entity embedding $g(e)$ before feeding into the final MLM prediction layer. By including entity embeddings in the prediction of related text tokens, we get them to "remember" important aspects from the text without sacrificing modularity.

3 Tasks

In this section, we define the three tasks used to evaluate pre-trained entity representations.

3.1 Supervised Task: Movielens Tag Prediction

The original MovieLens Tag Prediction task is to produce movie-tag scores for a set of movies and a canonical vocabulary of tags (see examples in Table 1), based on a collection of crowdsourced (movie, tag, user) votes, as well as (user, movie) star ratings. These tags are often not factual but may refer to plot elements, qualitative aspects or reflect subjective opinions. Since the same can be said about user reviews, and we observe a non-trivial amount of textual entailment between the two sources. We therefore intentionally exclude user ratings from the input. The new challenge is to complete the movie-tag relevance matrix by leveraging movie reviews, hereafter referred to as the *closed-vocabulary tag prediction* task[7]. This is a supervised setup where models are fine-tuneed with tag labels and evaluated on a held-out set subset of movies, as elaborated in Section 5.

3.2 Few-Shot Task: Open Vocabulary Tag Prediction

In reality, the space of tags is not static. Rather, tags are a useful kind of user-generated content that evolves to reflect the zeitgeist, much like human language. Many online platforms (e.g, Twitter and Instagram to name a few) have vibrant online communities that keep inventing new tags. We therefore propose a new *open-vocabulary* formulation of the tag prediction problem where any phrase is allowed to be a tag.

This requires a small change in evaluation. Instead of held-out movies, we hold out a subset of tags and fine-tune on the rest (and on all the movies). Note that this is no longer a classic multi-label classification task as we never get to see the test labels during training. Rather, this open-vocabulary setup is akin to answering boolean questions (about a movie) based on a text document (Clark et al., 2019).

3.3 Zero-Shot Task: Reddit Movie Suggestions

The purpose of this task is to evaluate pre-trained entity representations in the context of vertical search. The classic entity ranking problem is, given a text query and a finite set of entities, to rank them according to their relevance to the query. Recall that DocEnt models are naturally designed to make such relevance predictions via $P(Entity|Sentence)$ — without any fine-tuning, if necessary. We therefore leverage the Reddit Movie Suggestions Dataset (detailed in Section 4.3) as a source of both queries and ground truth to define a zero-shot movie ranking task. To clarify, the notion of *zero shot* implies a pre-trained but not fine-tuned model in our context. This dataset is particularly interesting for its challenging queries, with their distinctly natural, often conversational language (e.g., *"Last week I watched the British cold war movie Threads. I am scarred, but intrigued as well. Any similar deeply disturbing yet realistic movies you can recommend?"*, see Table 2 for more examples). Another challenge is an explicit recommendation intent present in many of the queries (i.e., *"Movies like ..."*), making this task a mixture of Search and Recommendation. The latter typically requires specialized recommendation models of entity-to-entity

[6] Conversely, a very large $\mathcal{E}$ may require an optimized implementation of softmax to maintain scalability.

[7] One can also view this as a two-dimensional knowledge base (KB) completion problem, where relation types are not available and the KB is reduced to a 2D matrix.

Query	Top 5 Results
Movies like [Whiplash] about an artist or a musician chasing an almost impossible dream and nearly or does ruin his life because of it	Inside Llewyn Davis, **Whiplash**, A Young Man with a Horn, Hustle & Flow, Born to Be Blue
Really dark, slow paced movies with minimal story, but incredible atmosphere, kinda like [Drive] or [The Rover]	**The Rover**, Valhalla Rising, Only God Forgives, Blade Runner, Sicario
Films like [Mission Impossible] or [The Italian Job] that have big scenes where the characters must break in or infiltrate some place	National Treasure: Book of Secrets, **Mission: Impossible – Rogue Nation**, Ant-Man, **The Italian Job**

Table 2: Qualitative examples illustrating zero-shot movie ranking by DOCENT-FULL, with natural language queries crawled from Reddit. The bracketed greyed-out movie mentions are users' examples of desired recommendations, removed from the queries to probe the model in what resembles a movie guessing game. Those obfuscated entities were correctly guessed by the model based on remaining query terms, making it to the Top 5 in most cases. Other top matches appear to be equally relevant.

similarity, and cannot generally be solved with keyword-based search.

4 Datasets

4.1 Amazon Movie Reviews Corpus

All our models are pretrained on Amazon Product Reviews (He and McAuley, 2016) in the "Movies and TV" category, comprising 4,607,047 reviews for 208,321 movies collected during 1996–2014[8].

4.2 Reviews2Movielens

One of this paper's contributions is *Reviews2Movielens*—a new multi-document multi-label dataset created by joining Amazon Movie Reviews (He and McAuley, 2016; Ni et al., 2019) and MovieLens (Harper and Konstan, 2016), a rich source of crowdsourced movie tags. The key challenge in joining the two datasets is establishing correspondences between their respective movie IDs, which turns out to be a many-to-one mapping[9]. We have identified a subset of high-precision many-to-one correspondences by applying Named Entity Recognition techniques[10] to both Amazon product titles (incl. release years) and their product pages. The resulting mapping consists of 71,077 unique Amazon IDs and 28,918 unique MovieLens IDs. The mapping accuracy was manually verified to be 97% based on 200 random samples. Ultimately, the joined dataset contains nearly 2 million reviews and close to 1B words, significantly more than its IMDB counterpart (Maas et al., 2011).

Since both datasets are widely used as a source of data and academic benchmarks (Miller et al., 2003; Jung, 2012; Anand and Naorem, 2016; He and McAuley, 2016; Ni et al., 2019), we hope that this new mapping[11] will be useful to the community.

4.3 Reddit Movie Suggestions

This user-generated dataset contains a collection of 4765 movie-seeking queries and corresponding recommendations, collectively curated and voted on by the Reddit Movie Suggestions community[12]. Worth noting are (a) the conversational, human-to-human language of the queries; (b) the community-recommended movies that, while sparse and possibly biased, can be used as a source of ground truth. While modest in size, the dataset is well-suited to evaluate zero-shot performance on the movie ranking task defined in Section 3.3.

5 Experiments

5.1 Pre-training

All our experiments start with pre-training models on the Amazon Movie Reviews corpus, followed by optional task-dependent fine-tuning. First, we apply some simple filtering to the input, removing reviews shorter than 5 words and movies with less than 5 reviews [13]. This results in 81,057 Amazon movies, of which 17,131 have MovieLens correspondences, and 4,181,727 reviews in total. Further, we split reviews into individual sentences (or short paragraphs) so as to circumvent the BERT

[8] We've used the 2016 version of the dataset from http://jmcauley.ucsd.edu/data/amazon.

[9] Each Amazon ID (ASIN) matches a canonical product URL, e.g., https://www.amazon.com/dp/B06XGG4FFD. However, these IDs correspond to specific product *editions* (typically DVDs) rather than unique titles, causing duplication issues. Some are collections of several titles.

[10] We use the public Google Cloud Natural Language API – https://cloud.google.com/natural-language/docs/basics#entity%20analysis.

[11] See http://goo.gle/research-docent

[12] https://www.reddit.com/r/MovieSuggestions

[13] This low-count filtering is applied after de-duplication and aggregation.

sequence length limit. Finally, since our goal is to learn non-obvious entity attributes, we remove movie names from their reviews.

All our models use the standard BERT-base configuration with 12 layers, 12 attention heads and a hidden size of 768, and are initialized with a publicly available BERT-base checkpoint[14].

5.2 Tag Prediction: Fine-tuning Strategies

We will now describe the fine-tuning strategies used to transfer pre-trained DOCENT models to downstream tag prediction tasks.

DOCENT-FULL To generate movie-tag relevance scores, we need to predict $P(Tag|Movie)$, which we cast as binary classification. Recall that BERT has a built-in binary classifier (for next-sentence prediction), implemented as a single-layer FFN [15] on top of its *[CLS]* output, with logistic loss. We simply repurpose that layer for our task.

DOCENT-DUAL and DOCENT-HYBRID Recall that, during pre-training, DOCENT-DUAL and DOCENT-HYBRID use softmax cross entropy loss to predict $P(Entity|Sentence)$. However, tag prediction poses the inverse problem: predict tags based on a movie entity. In our dual encoder framework, that can be done simply by computing softmax over all of the encoded tags rather than entities, without any changes to the architecture.

Shared Strategies For fine-tuning, all of the models share the following choices. First, we treat every existing movie-tag pair in the training set as a positive example, weighted proportionally to the number of user votes for that pair (or to the logarithm thereof). Next, for a given movie, about 10% of all vocabulary tags are sampled as negative examples, excluding the known true positives for that movie. To prevent overfitting, we fix entity embedding weights for all models during fine-tuning.

5.3 Entity-less Baselines

To corroborate the utility of explicit entity representations, we set out to evaluate a few baselines that circumvent them by representing each entity as a Bag-of-Sentences (BoS), computed over its related reviews with a sentence encoder of choice. Such a BoS encoder can replace entity embeddings in our architecture, yielding a naïve variant of DOCENT-DUAL. We call these baselines BOS-GLOVE, BOS-BERT and BOS-SENTENCEBERT[16], reflecting their underlying sentence encoders.

Task	Movies	Tags	M-T Pairs
Closed (test)	1000	1128	46359
Closed (dev)	380	1128	17943
Open (test)	6392	500	141618
Open (dev)	3362	100	25274

Table 3: Evaluation datasets sizes for Tag Prediction tasks. Closed / Open stand for the closed and open vocabulary tasks, respectively; M-T Pairs shows the number of corresponding movie-tags pairs. The top two rows describe *movie* holdout sets used in our closed vocabulary experiments; bottom two rows showing *tag* holdouts for open vocabulary experiments.

5.4 Evaluation

5.4.1 Movielens Tag Prediction

The main challenge with evaluating tag prediction is the sparse and noisy nature of user-generated ground truth. For instance, a certain movie tag having zero votes may still be relevant in reality. On the other hand, some entities may have votes for contradictory tags (e.g., both "funny" and "not funny"). The original Tag Genome baseline (Vig et al., 2012b) mitigated this by collecting an additional dataset of unbiased movie-tag relevance scores. Alas, that data has not been released. Instead, we propose two complementary metrics that cast tag prediction either as binary classification or as a ranking problem.

For classification, we binarize labels as follows. Let $\#(m,t)$ be the number of users who assigned a tag t to a movie m. Then its binary counterpart $l(m,t)$ is set to 1 iff $\#(m,t) > T$, a threshold[17].

For the tag ranking formulation, we make the assumption that true movie-tag relevance is correlated with the number of movie-tag votes, and define our movie-tag relevance score as $r(m,t) = \#(m,t)$.

Equipped with this score, we use Precision@k and NDCG metrics (Järvelin and Kekäläinen, 2002) to measure performance.

Tag prediction baselines include

MovielensTopTags— a fixed ordering of tags.

[14] https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

[15] Feed-Forward Neural Network

[16] SENTENCEBERT (Reimers and Gurevych, 2019) fine-tunes BERT on NLI to provide off-the-shelf semantic sentence representations.

[17] We use $T = 2$ to filter out noisy tags.

Model	MAP	AUC
MovielensTopTags	6.2	0.80
TD-IDF	32.3	0.86
BOS-BERT	39.3	0.91
TAGGENOME	43.9	**0.98**
DOCENT-FULL	**44.7**	**0.98**
DOCENT-DUAL	38.6	0.96
DOCENT-HYBRID	44.1	**0.98**
Human	76.6	0.99

Table 4: Mean Average Precision and ROC-AUC results on the closed-vocabulary tag prediction task. TAGGENOME is the original baseline from MovieLens creators (Vig et al., 2012b), trained on multiple additional features and considered SOTA. Despite using fewer features, DOCENT matches TAGGENOME performance on AUC and outperforms it on precision (MAP).

Model	MRR	Recall, %	
		@50	@100
Lucene (TF-IDF)	0.14	15.3	20.7
BOS-GLOVE	0.04	4.1	6.6
BOS-BERT*	0.08	9.6	14.2
BOS-SENTENCEBERT	0.07	7.6	11.7
DOCENT-FULL	0.22	21.3	28.4
DOCENT-DUAL	0.27	28.0	36.3
DOCENT-HYBRID	**0.31**	**31.9**	**40.9**

Table 5: Zero-shot results for DOCENT models vs several baselines on Reddit Movie Suggestions. MRR stands for Mean Reciprocal Rank.

TF-IDF scores for movie-tag pairs, based on tag frequencies in movie reviews.

BOS-BERT, as defined in Sec. 5.3, is fine-tuned to estimate sentence-to-tag relevance directly[18]. This setup is applicable to both open and closed vocabulary scenarios. During inference, a movie-tag prediction is obtained by averaging over sentence-wise predictions for the movie's reviews.

TAGGENOME—the original baseline from MovieLens team (Vig et al., 2012b). The comparison is not entirely apt as that model was trained on additional movie-tag relevance data and user ratings, albeit with a smaller corpus of unsupervised reviews. Also, TAGGENOME was trained on all of MovieLens (no holdouts).

Humans—to simulate human performance, apply cross-validation to ground truth user votes, treating one of the folds as a quasi-model.

All models were evaluated on the same holdout sets, with averaging.

Closed Vocabulary Tag Prediction In this scenario, evaluation is done on a holdout set of movies (with a smaller development set used for hyperparameter tuning; see Table 3 for details).

Results for ranking (MAP) and binary classification (AUC) metrics are shown in Table 4. Collectively, DOCENT models outperform the strong TAGGENOME baseline on tag ranking (see also Fig. 2 (a) and (b)) and match (or slightly outperform) it in binary classification. It is a strong result, considering that DOCENT had no access to additional features used by TAGGENOME and employed no feature engineering. Of the three models, DOCENT-DUAL scores the lowest on all metrics, likely due to not optimizing for $P(Text \mid Entity)$ in pre-training. Finally, note that all models still score way below humans on the (harder) tag ranking task, indicating considerable headroom.

Open Vocabulary Tag Prediction This task is evaluated by withholding parts of the tag vocabulary so that those tags are never seen in training (consult Table 3 for details). Fig. 2 (c) shows our models' performance on the binary classification task base on the fraction of the vocabulary seen by a model in fine-tuning. The graph shows that training with only 100 of the 1124 tags results in reasonable performance. Of our three models, DOCENT-FULL starts below the others but adapts the fastest, reaching a near-closed vocabulary performance with less than 50% of the full tag vocabulary.

5.4.2 Reddit Movie Suggestions

Movie suggestion baselines Since this is a search task, we compare our models to an Apache Lucene[19] baseline, arguably the world's most widely used open-source search engine. For completeness, we also compare to BOS-BERT*[20], BOS-GLOVE and BOS-SENTENCEBERT, neural baselines defined in Sec. 5.3, whose query-movie relevance score is given by the maximum cosine similarity among the movie's review sentences[21].

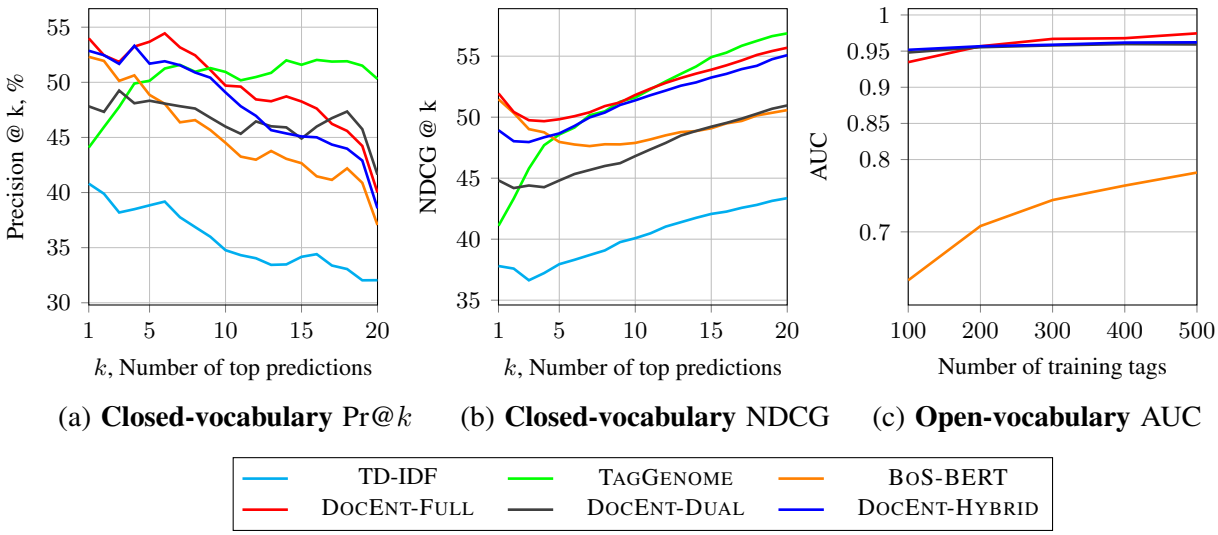
Table 5 shows the Mean Reciprocal Rank (MRR) as well as recall, metrics that suit the noisy ground

[18] We found it is best to encode a review sentence using BERT's *[CLS]* output, while tags are encoded by averaging individual tokens' output vectors.

[19] https://lucene.apache.org/

[20] In absence of a fine-tuned *[CLS]* output, this version of BOS-BERT encodes sentences by averaging their individual tokens' output vectors.

[21] In this case, we found that aggregating sentence-wise predictions with L^∞ norm is superior to averaging.



(a) **Closed-vocabulary** Pr@k (b) **Closed-vocabulary** NDCG (c) **Open-vocabulary** AUC

Figure 2: Performance on tag prediction tasks. Left and center: Precision and NDCG @k, with a closed vocabulary. DOCENT-FULL dominates the strong TAGGENOME baseline for smaller values of k, a concentration of gains typical for binary classification models. For perspective, human Precision@k ranges 80-95% for this task. Right: AUC for open vocabulary experiments, with models trained using a variable fraction of the tag vocabulary. DOCENT approaches close-vocabulary AUC after training with only 10-50% of the vocabulary (showing all baselines that were available to us in this setting).

truth (for completeness, see also the qualitative results in Table 2). DOCENT models outperform the Lucene baseline on all metrics, with DOCENT-HYBRID leading by a large margin. Compared to DOCENT-DUAL, its strong performance is not surprising since DOCENT-HYBRID optimizes both $P(Entity \mid Text)$ and $P(Text \mid Entity)$—a combination of tasks that helps avoid overfitting.

Also expected is the relatively weak performance of DOCENT-FULL. As discussed in Sec. 2, its high-capacity entity representations are prone to overfitting when the number of entities is relatively small. Still, this shortcoming can be remedied by fine-tuning, as evidenced by this model's superior results on tag prediction in Sec. 5.4.1. These results suggest that DOCENT-FULL may be a good choice in semi-supervised scenarios.

6 Related Work

Much of the prior art in text-based entity understanding is motivated by the *Entity Linking* (EL) problem: predict a unique entity from its mention in text, assuming a single right answer. By contrast, tasks like entity retrieval and tag prediction imply multiple valid matches and emphasize understanding entities through the prism of their attributes, expressed in natural language. Still, recent EL works propose dual encoder approaches similar to ours (Yamada et al., 2017; Ling et al., 2020; Cheng and Roth, 2013; Sun et al., 2015; Yamada et al., 2016; Chang et al., 2020; Kobayashi et al., 2016; He et al., 2013; Gupta et al., 2017), with Ling et al. (2020) already discussed in Section 2.1. Dual encoders have also been explored in zero-shot scenarios (Gillick et al., 2019; Logeswaran et al., 2019; Wu et al., 2019; Gupta et al., 2017), with entity embeddings computed dynamically based on metadata such as dictionary definitions, entity name and/or category. Others incorporate entity representations directly in the transformer by retrieving from an external memory (Févry et al., 2020; Peters et al., 2019). While clearly useful for EL, e.g., in sentences with multiple entity mentions, the benefits to our applications are unclear. Finally, there is ERNIE (Sun et al., 2019) – a language model trained with awareness of entity mentions. Alas, the lack of explicit entity representation limits its use in our tasks.

7 Conclusion & Future Work

This paper proposes a family of models to learn self-supervised entity representations from large document collections. We motivate these dedicated representations by contrasting them with naive text-as-a-proxy approaches, with clear gains on entity-centric tasks such as natural language search and movie tag prediction. We then show that achieving superior performance requires optimizing both $P(Entity \mid Text)$ and $P(Text \mid Entity)$—in contrast to the baseline RELIC model (and similar

prior dual encoders) having only a single objective. To that end, we propose two novel models and study them in zero-shot, few-shot and supervised settings. We match or outperform competitive baselines, where available, with little or no fine-tuning.

Future Work As shown qualitatively in Sec. 3.3, DOCENT has the potential for being a hybrid approach to bridge entity retrieval and recommendation, an application worth exploring in depth (e.g., on the MovieLens Recommendation task which can be readily integrated with DOCENT thanks to *Reviews2Movielens*). A larger entity retrieval study with heterogeneous entity types is another useful direction. Lastly, extending DOCENT to additional entity understanding tasks such as QA and summarization is yet another promising avenue.

Acknowlegements

We appreciate the feedback from the reviewers. This work is partially supported by NSF Awards IIS-1513966/ 1632803/1833137, CCF-1139148, DARPA Awards#: FA8750-18-2-0117, FA8750-19-1-0504, DARPA-D3M - Award UCB-00009528, Google Research Awards, gifts from Facebook and Netflix, and ARO# W911NF-12-1-0241 and W911NF-15-1-0484.

References

Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM.

Deepa Anand and Deepan Naorem. 2016. Semi-supervised aspect based sentiment analysis for movies using review filtering. *Procedia Computer Science*, 84:86–93.

Jill Burstein, Christy Doran, and Thamar Solorio, editors. 2019. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, Washington, USA. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In (Burstein et al., 2019), pages 2924–2936.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In (Burstein et al., 2019), pages 4171–4186.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego García-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 528–537. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2681–2690. Association for Computational Linguistics.

F. Maxwell Harper and Joseph A. Konstan. 2016. The movielens datasets: History and context. *TiiS*, 5(4):19:1–19:19.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 30–34. The Association for Computer Linguistics.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Jason J Jung. 2012. Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. *Expert Systems with Applications*, 39(4):4049–4054.

Sosuke Kobayashi, Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2016. Dynamic entity representation with max-pooling improves machine reading. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 850–855. The Association for Computational Linguistics.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37.

Jeffrey Ling, Nicholas FitzGerald, Zifei Shan, Livio Baldini Soares, Thibault Févry, David Weiss, and Tom Kwiatkowski. 2020. Learning cross-context entity representations from text. *CoRR*, abs/2001.03765.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3449–3460.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6294–6305.

Bradley N Miller, Istvan Albert, Shyong K Lam, Joseph A Konstan, and John Riedl. 2003. Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266. ACM.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations.

Matthew E. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1333–1339. AAAI Press.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Jesse Vig, Shilad Sen, and John Riedl. 2012a. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):13.

Jesse Vig, Shilad Sen, and John Riedl. 2012b. The tag genome: Encoding community knowledge to support novel interaction. *TiiS*, 2(3):13:1–13:44.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany. Association for Computational Linguistics.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning distributed representations of texts and entities from knowledge base. *Trans. Assoc. Comput. Linguistics*, 5:397–411.