New or Old? Exploring How Pre-Trained Language Models Represent Discourse Entities

Sharid Loáiciga[*1], **Anne Beyer**[*2], **David Schlangen**[2]
[1]CLASP, Dept. of Philosophy, Linguistics and Theory of Science, University of Gothenburg
[2]Computational Linguistics, Department of Linguistics
University of Potsdam, Germany
sharid.loaiciga@gu.se
anne.beyer,david.schlangen@uni-potsdam.de

Abstract

Recent research shows that pre-trained language models, built to generate text conditioned on some context, learn to encode syntactic knowledge to a certain degree. This has motivated researchers to move beyond the sentence-level and look into their ability to encode less studied discourse-level phenomena. In this paper, we add to the body of probing research by investigating discourse entity representations in large pre-trained language models in English. Motivated by early theories of discourse and key pieces of previous work, we focus on the information-status of entities as discourse-new or discourse-old. We present two probing models, one based on binary classification and another one on sequence labeling. The results of our experiments show that pre-trained language models do encode information on whether an entity has been introduced before or not in the discourse. However, this information alone is not sufficient to find the entities in a discourse, opening up interesting questions about the definition of entities for future work.

1 Introduction

In a seminal paper from 1969, Karttunen imagines "a device designed to read a text in some natural language, interpret it, and store the content in some manner, say, for the purpose of being able to answer questions about it". Such a device—considered by him "not a practical idea, for the time being at least"—he says would need to have a particular feature, namely that it "be able to recognize when a novel individual is mentioned in the input text and to store it along with its characterization for future reference."

Now, more than 50 years later, neural models appear to have made such a device a practical idea after all. But do they recognize when a text introduces a new entity into the "universe of discourse", or when, in contrast, the new information concerns a previously introduced one? This is the question that we are asking in this paper.

Following Karttunen's idea and inspired by Prince (1992)'s analysis of information-status, we focus on discourse entities. In particular, we target the task of distinguishing between the status of entity mentions as discourse-new or discourse-old. Considering that discourse entities are central to discourse theories and meaning, we consider that this is an understudied subject in the field. We take a step back from much more specific tasks such as coreference resolution and look at entities being referred to over time. We believe that the new/old distinction, as a simplified form of discourse representation, lets us ask whether language models are able to *keep track* of discourse entities.

Concretely, we build probing models that take as input the representations of pre-trained English language models and predict discourse-new/-old values for all mentions in a text. We present two probing models tackling the task on two different levels of complexity: binary classification and sequence labeling. The first probe takes one entity mention and its preceding context up to that point, and produces a binary decision (discourse-new/-old). It tells us to what extent the context matters for this task. Inspired by the Named Entity Recognition task, the second probe labels each token in a sequence (new/old/outside). It tells us thus whether the entities can be localized in the sequence and labeled with the correct type. A few pieces of research have found first indications about the presence of entity knowledge in pre-trained language models. In particular, Sorodoc et al. (2020) focus on identifying pronoun-antecedent pairs, leaving the question open of whether models have a general notion of entities. Li et al. (2021) work only with synthetic data, which is simple in nature with short sentences and few entities. Last, Gupta and Durrett (2019b) find that pre-trained language model's representations are unable to trace explicit entity state

[*] Shared first authorship.

Proceedings of the 29th International Conference on Computational Linguistic, pages 875–886
October 12–17, 2022.

changes in recipes and physical processes.

Our findings suggest that contextualized pre-trained language model representations generated with a transformer model contain enough discourse information to determine whether an entity is new or old—with results as high as 0.89 F1 in the classification probe, even beyond the case of pronouns. However, that knowledge does not suffice to localize the entity in the sequence—with results as low as 0.51 F1 in the sequence labeling task. [1]

2 Related Work

The intuition that language models implicitly capture and in turn also benefit from entity knowledge has been explored for some time now (Ji et al., 2017; Yang et al., 2017; Schuster and Linzen, 2022, *inter alia*), with recent papers focusing on how to inject some explicit entity representation into the system (Aina et al., 2019; Gupta and Durrett, 2019a, among others). The information-status distinguishes between discourse entities that are newly introduced in the text and those that are already known to the comprehender (Prince, 1992; Kamp and Reyle, 1993). It is a central part in discourse theories as it accounts for the changes in referring expressions used to re-mention discourse entities as they undergo meaning updates as the context evolves.

Our work is most similar to that of Sorodoc et al. (2020) and Li et al. (2021). Both of these papers are interested in probing entity knowledge in pre-trained language models at the discourse level, and they both take a semantic approach in that they are interested in the similarity between different mentions of the same entity at different points in the discourse.

Working with the OntoNotes (Pradhan et al., 2012) coreference corpus, Sorodoc et al. test whether pre-trained language model representations have the morpho-syntactic and semantic knowledge required to match a pronoun with its antecedent. They report results based on pre-trained representations generated with both a Transformer and an LSTM model. Using two baselines i) always referring to the nearest mention, ii) always referring to the most similar (cosine similarity) token, they found that a probe fed with the pre-trained embeddings succeeds at the task of predicting the correct antecedent (75.9% accuracy). An error analysis of the probe with the Transformer representations showed that noun phrases were harder to solve than pronouns (so they focus on the latter). The probe also succeeded in learning agreement, as tested by inserting distractors, but accuracy drops to 53% in hard cases (e.g., when the pronoun and antecedent disagree in gender/number). In our experiments, we go a step further and consider pronouns as well as noun phrase mentions.

Focusing on Transformers, Li et al., on their side, work with the Alchemy (derived from Long et al. (2016)) and Textworld (Côté et al., 2019) datasets. They use the data to construct logical propositions which are then classified into True/False with a binary classifier probe (e.g., *You see an open chest. The only thing in the chest is an old key. The chest contains an apple.* → *True/False*). This data transformation is possible because the original data are constructed short documents with simple sentences and few entities. It should be noted as well that although the probe itself is a low-capacity linear classifier, it needs a proposition embedder and a localizer of the entity in the sequence as additional pipeline components. Their results are measured through accuracy (the aggregation of all propositions with an entity) and they go as high as 94%. Our work is concerned with real world text data instead. As this increases the space of possible propositions, it also requires us to simplify the task, which we will explain in detail in the next sections.

Turning to the approach of probing or diagnostic classifiers (Hewitt and Liang, 2019), these are simple systems trained on the encoded representations of another system. If the probe succeeds in the task for which it is trained—discourse-new/discourse-old in our case, we conclude that the input had the necessary knowledge to solve the task. Research based on probing models mostly relies on classification tasks. This paper is a part of that, but here we additionally use a sequence labeling task. This strategy has precedent in examples such as Ramponi et al. (2020) and Dai et al. (2019a) who use sequence labeling for event and entity extraction, respectively. In the context of probing discourse knowledge, Koto et al. (2021) has used this format for a sentence ranking experiment where the probe was asked to predict the most likely sentence ordering as a sequence.

[1] The code for our experiments is available at: https://github.com/clp-research/new-old-discourse-entities.

(1) [The researchers]$_t$ said [they]$_t$ have isolated [a plant gene that prevents [the production of [pollen]$_i$]$_j$]$_m$. [The gene]$_m$ thus can prevent [a plant]$_y$ from fertilizing [itself]$_y$.

(2) The researchers said they have isolated **[a plant gene that prevents the production of pollen]**. – **new**
The researchers said they have isolated a plant gene that prevents the production of pollen. **[The gene]** – **old**

(3) The researchers said they have isolated a plant gene that prevents the production of **[pollen]**. – **new**
The researchers said they have isolated a plant gene that prevents the production of pollen. The **[gene]** – **old**

(4) [The researchers] said [they] have isolated [a plant gene that prevents the production of pollen] . [The gene] thus can prevent [a plant] from fertilizing [itself] .
B-old I-old O B-old O O B-new I-new I-new I-new I-new I-new I-new I-new I-new . B-old I-old O O O B-new I-new O O B-old .

(5) The [researchers] said [they] have isolated a plant [gene] that prevents the [production] of [pollen] . The [gene] thus can prevent a [plant] from fertilizing [itself] .
O B-old O B-old O O O O B-new O O O B-new O B-new . O B-old O O O O B-new O O B-old .

Figure 1: Excerpt from one document extracted from the ARRAU corpus. (1) presents the original entities with their embedded mentions, (2) shows the spans data for the classification experiments, while (3) the heads data, (4) shows our formatting into IOB labels for the sequence labeling experiments for spans, and (5) shows our formatting into IOB labels for heads (see Sec. 4.2).

3 Data and Pre-trained Model

We present two sets of probing experiments on the hidden representations of a pre-trained model using real-world text data.

3.1 Data

For our experiments, we use data from the ARRAU corpus (Uryupina et al., 2020), a richly annotated coreference corpus which includes information-status (discourse-new/-old) annotations.[2] We only consider referring mentions, as non-referring mentions do not introduce a discourse entity and thus are not annotated with their information-status. The ARRAU corpus contains annotations for all NPs, including singletons (discourse-new entities that do not occur as antecedents for any other mention). We kept the official split used in the CRAC shared task[3] (Poesio et al., 2018), but discarded the spoken dialog and narrative genres (`trains` and `pears`, respectively) because they are presumably different from the domain of the pre-trained model. We construct two data variants, *spans* and *heads*. The examples per partition can be found in Table 1.

Spans As Fig. 1 shows, the original ARRAU data contains several embedded entities (1), that require some adaptation. We first remove all embedded entities from the maximal entity span containing them. For example, the mentions *i* and *j* have been removed from the outer entity span *m*, *a plant gene that prevents the production of pollen* (4). If an embedded mention in the current sentence is re-mentioned in a subsequent sentence, the remention is retained (despite the first embedded mention not being retained). Then, for the sequence labeling task, we create standard IOB labels: an entity's first token is either B-NEW or B-OLD and subsequent tokens have an I-NEW or I-OLD label. Non-entity tokens are labeled as outside (O).

We expect this version of the data to be easier for the classification probe as it has access to all the information in the span, but harder for the sequence labeling probe, since there is much variability in the length of the mentions. Identifying the span also implies higher order knowledge of the notion of entity. Neural coreference resolution systems, for instance, explicitly learn the representation of spans as the combination of the vectors for the first and last tokens and the head of a mention (Lee et al., 2017).

Heads In this version, we take only the head of each mention. This means that the probing model only needs to label the lexical head or pronoun, as opposed to the complete span of a mention. We obtained the heads by extracting the *min_words* attribute from the manual ARRAU annotations.[4]

[2]The complete corpus is available through the LDC: https://catalog.ldc.upenn.edu/LDC2013T22. Some parts are also available through the ARRAU corpus GitHub, according to the authors: https://sites.google.com/view/arrau/corpus.

[3]http://anawiki.essex.ac.uk/dali/crac18/crac18_shared_task.html

[4]This can either be a pronoun, head noun or proper name.

	Heads			Spans		
	Train	Dev	Test	Train	Dev	Test
New	29,117	1,991	5,084	16,639	1,141	2,812
Old	19,171	1,391	3,672	9,814	714	1,883
Total	48,288	3,382	8,756	26,453	1,855	4,695

Table 1: Data splits for the discourse entity probes.

Here, embedded entities are retained.

3.2 Pre-trained Representations

All of our probes are based on the representations learned by a pre-trained Transformer-XL model (Dai et al., 2019b) (available through the Hugging Face library (Wolf et al., 2020) as TRANSFO-XL-WT103). We focus on this English model first, as it has been used in the closely related work by Sorodoc et al. (2020) and because it is explicitly able to capture long contexts and generate "relatively" coherent long texts (Dai et al., 2019b) by using a recurrence mechanism over cached previous segment states. This counteracts the "context fragmentation" introduced by chopping off contexts at a given length to cope with limited computational capacities.

We extract the last 1024-dimensional hidden state representations for each token by feeding the pre-trained model a whole document at a time, so they are contextualized with the discourse knowledge encoded by the model. In Section 4.3, we show how to extend our probes to other models and compare the results to pre-trained representations extracted from GPT-2 (Radford et al., 2019), which is not specifically adjusted for longer inputs.[5].

3.2.1 Baselines

To interpret the results of our probing models, we compare them to models initialized with static 300-dimensional fastText embeddings (Grave et al., 2018), which we extract word by word using the Python `fastText` module.[6] We further match the results against two simple baselines, a majority baseline that labels every entity as new, and one based on POS tags where only entity mentions that start with a definite article or pronouns are considered old. The intuition behind the second one is that these are easy and frequent cases, which may reveal whether our models simply rely on these linguistic cues.

[5]An extension to different languages, however, would also require gold data with information-status annotations and is not included in this work.

[6]https://fasttext.cc/docs/en/python-module.html

4 Probing Experiments

We first perform a classification task to gather information about the entity representations themselves. The second probing task looks at sequence labeling in order to evaluate whether this information can be also used to detect entity boundaries.

4.1 Classification Task

Previous work on entity status tracking (Gupta and Durrett, 2019b) framed the task as entity classification by pre-extracting the entity in question, thereby not requiring the model to identify what an entity is in the first place. We adapt this task in order to probe whether pre-trained hidden representations contain information about whether an entity is newly introduced into the discourse or if it is a re-mention of an already introduced entity.

4.1.1 Pre-processing

First, we split each text incrementally at each entity mention such that the context contains the n words up to the first entity (and in the next sample the second and so on), and the target contains the respective following entity tokens. As described in Sec. 3.1 we extract either only the heads or the maximal spans of entities as the target. We prepend the $\langle eos \rangle$ token to the context to avoid empty contexts when the first token is (part of) an entity (e.g., in the very first sentence of a document).

Next, we use the pre-trained Transformer-XL representations for every context token, and sum over the extracted target entity vectors to get an entity representation (see lower part of Fig. 2).

4.1.2 Models

Following the baseline models of Gupta and Durrett (2019b), we train an attention-based classifier that computes bilinear attention (Luong et al., 2015) between the entity representation and the context tokens and predicts the entity category (new/old) from the combined result. We use the Attention implementation from `pytorch-nlp` (Petrochuk, 2018) (Eq. 1-4) and a linear layer with a sigmoid function on top (Eq. 5).

$$a_i = h_{c_i}^T * W_{att} * e \quad (1)$$

$$\alpha = softmax(a) \quad (2)$$

$$context = \sum \alpha_i * h_{c_i} \quad (3)$$

$$h_{c,e} = tanh(W_{comb} * [context, e]) \quad (4)$$

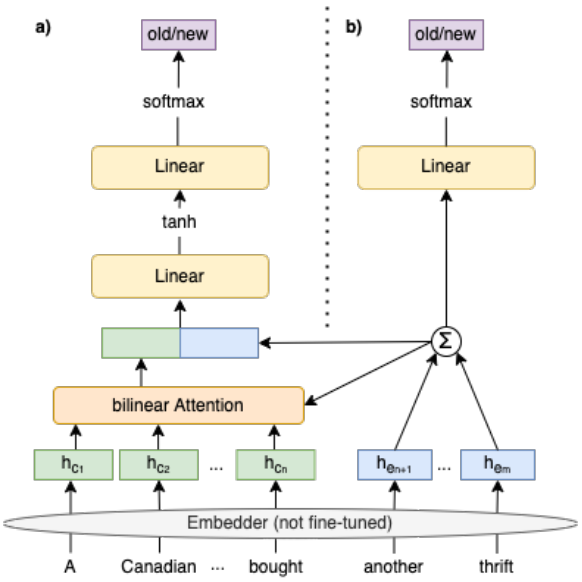

Figure 2: Probing classifier architectures. Embeddings are previously extracted document-wise from the hidden layer of a pre-trained Transformer-XL model (*Embedder*). a) Contextualized classification based on attention between context ($c_1...c_n$) and summed entity ($e_{n+1}..e_m$) representation (Eq. 1-5). b) Entity classification based on summed entity ($e_{n+1}..e_m$) representation alone (Eq. 6).

$$P(y|h_{c_1}, ..., h_{c_n}, e) = sigmoid(W_a * h_{c,e} + b) \quad (5)$$

This model has access to the whole context up to the entity. To gather further insights on the role of the context tokens in terms of what kind of information is already encoded in the entity representation itself, we additionally train a model without context, using only the entity representation to predict its status (Eq. 6).

$$P(y|e) = sigmoid(W_b * e + b) \quad (6)$$

The model architectures are displayed in Fig. 2. Training details and hyperparameters used are provided in Appendix A.

4.1.3 Results

The results of the classification experiments are displayed in Table 2. While the fastText embeddings already yield an improvement over the majority and the POS baseline, the Transformer-XL embeddings yield the best overall results, suggesting that the contextualization adds some useful information on the entity state. Surprisingly, however, there is not much difference between the attention-based and the entity-based models, suggesting that the information required for this task is contained in the pre-trained representations themselves. It is also interesting that taking the whole span into account improves the results for the fastText embeddings, but yields no gain for the Transformer-XL representations. This suggests that due to the contextualization, the necessary information is already encoded in the head representation itself. A detailed discussion follows in Sec. 5.

4.2 Sequence Labeling Task

Inspired by the NER scenario, our second probing model takes the form of a sequence labeling task, whereby each token in the sequence is assigned a categorical label. Discourse entities are a broader category of named entities, so instead of assigning entities a type (e.g., ORGANIZATION, PERSON, TIME, etc.), the probe assigns *new* or *old* labels, following the IOB scheme. An example is shown in Fig. 1.

This framework offers us two advantages in a single task: i) the probe has to localize the entity in the sequence (an additional pre-processing step in previous work), and ii) the probe has to assign a classification label to the entity.

4.2.1 Pre-processing

We slice the extracted hidden vector sequence according to the tokens in the original sentences, in order to feed our probes with examples at the sentence-level. In other words, the hidden representations are extracted based on the whole document, but the probing model labels the document sentence by sentence. The fact that our probes work at the sentence-level can be seen as a safety switch that limits their power, so no further contextualization– beyond that from the original embeddings– occurs. Because this precludes the probe from accessing any embeddings beyond the sentence, any success at predicting discourse-new or discourse-old should come from the entity embeddings themselves. This has the added advantage of easing the computation cost.

4.2.2 Model

The underlying method for our probe is a linear chain conditional random field (CRF) model.[7] The input to the model are sequences of n vectors, where each input h_i is a contextualized vector with size 1024 yielded by the pre-trained language model. This sequence of pre-trained vectors is fed

[7] We took advantage of the freely available implementation at https://github.com/kmkurn/pytorch-crf

	Heads							Spans						
	Discourse New			Discourse Old			Acc.	Discourse New			Discourse Old			Acc.
	Prec.	Rec.	F1	Prec.	Rec.	F1		Prec.	Rec.	F1	Prec.	Rec.	F1	
Probing Transformer-XL														
Attention-based	0.86	0.92	**0.89**	**0.88**	0.80	**0.84**	**0.87**	**0.88**	0.91	**0.89**	**0.86**	**0.81**	**0.83**	**0.87**
Entity-based	**0.87**	0.91	**0.89**	0.87	**0.81**	**0.84**	**0.87**	0.85	**0.92**	0.88	0.86	0.76	0.80	0.85
Baselines fastText 300														
Attention-based	0.76	0.86	0.81	0.76	0.62	0.68	0.76	0.82	0.89	0.85	0.81	0.71	0.75	0.82
Entity-based	0.70	**0.93**	0.80	0.82	0.46	0.59	0.73	0.76	**0.92**	0.83	0.82	0.56	0.67	0.78
Baselines w/o embeddings														
POS-based	0.66	0.83	0.73	0.63	0.40	0.49	0.65	0.74	0.80	0.77	0.66	0.57	0.61	0.71
Majority class	0.58	1.00	0.73	0.00	0.00	0.00	0.58	0.60	1.00	0.75	0.00	0.00	0.00	0.60

Table 2: Average results from five different random seeds of discourse-new vs. discourse-old classification experiments, probing pre-trained Transformer-XL representations versus static fastText embeddings (standard deviation is between 0.00 and 0.04 for all versions), and a POS-based (pronouns and defNP = discourse-old) and majority class (discourse-new) baseline

into an LSTM layer (Eq. 7) and a rectified linear unit activation function (Eq. 8), before being resized (Eq. 9) in order to fit the CRF (Eq. 10). The CRF layer finds the best possible sequence of labels $(y_i, ..., y_n)$ for the entire input sequence.

$$r_i = LSTM(h_i, r_{i-1}) \quad (7)$$

$$o_i = RELU(r_i) \quad (8)$$

$$l_i = W * o_i + b \quad (9)$$

$$p(y_1, ..., y_n | h_1, ..., h_n) = CRF(l_1, ..., l_n) \quad (10)$$

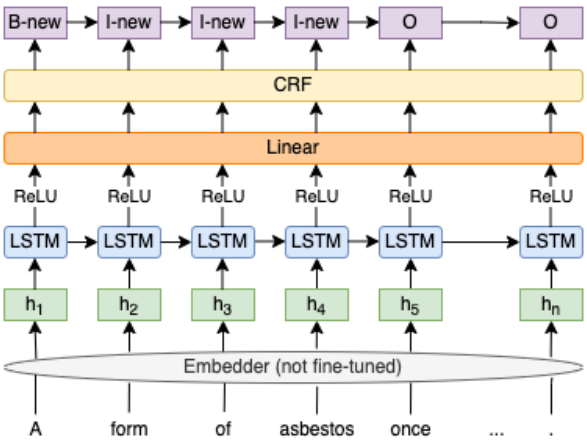

Figure 3: Sequence labeling model. The input to the probe are pre-trained representations h_i from the Transformer-XL model. After the LSTM layer (Eq. 7-8), a linear layer (Eq. 9) is needed to reduce the dimensions of the LSTM output from the hidden size to label size required by the CRF (Eq. 10). The CRF has a choice among 5 labels at each time step.

We present experiments with and without the LSTM layer. We expect a division of labor whereby the CRF learns from the syntactic signal (i.e., I comes after a B), and the LSTM learns the semantic content (i.e., new vs old). Training details and hyperparameters used are given in Appendix A.

4.2.3 Baselines

We build several model versions to estimate the success of our probes: In addition to the majority and POS baselines (cf. Sec. 3.2.1), we build a simple CRF using the Scikit-learn (Pedregosa et al., 2011) compatible CRFsuite (Okazaki, 2007) wrapper (Korobov, 2015) based on simple surface form features. These include whether a token is at the beginning or end of a sequence, whether the token starts with a capital letter, the last three characters of the token, and the last two characters of a token. We also add two versions of our probing models that do not rely on pre-trained representations but train the embeddings from scratch with dimensions 1024 for the comparison to Transformer-XL and 300 for fastText.

We do not compare to a human baseline for two reasons. First, we rely on human annotations of very high quality, for which annotators were asked to identify discourse entities as new/old before being asked to identify antecedents (Uryupina et al., 2020). Second, our main interest is to evaluate whether the pre-trained representations contain information that improves the performance on this task compared to our automatic baselines.

4.2.4 Results

The results for all configurations are reported in Table 3. For computing the scores, we used the SeqEval package (Ramshaw and Marcus, 1995; Nakayama, 2018).

A very clear pattern emerges: Heads are easier to identify than Spans, and discourse-old is easier to predict than discourse-new. It follows that the

	Heads							Spans						
	Discourse New			Discourse Old			Avg.F1	Discourse New			Discourse Old			Avg.F1
	Prec.	Rec.	F1	Prec.	Rec.	F1		Prec.	Rec.	F1	Prec.	Rec.	F1	
Probing Transformer-XL														
LSTM + Linear + CRF	**0.75**	0.79	**0.77**	**0.80**	**0.78**	**0.79**	**0.78**	0.59	0.59	0.59	**0.80**	**0.72**	**0.75**	0.66
Linear + CRF	0.70	0.70	0.70	0.75	0.69	0.72	0.71	0.43	0.38	0.41	0.69	0.63	0.66	0.51
LSTM + Linear + $\text{CRF}_{scratch}$	0.59	0.71	0.64	0.74	0.54	0.62	0.63	0.38	0.39	0.38	0.70	0.52	0.59	0.47
Linear + $\text{CRF}_{scratch}$	0.51	0.59	0.55	0.63	0.47	0.53	0.54	0.27	0.25	0.26	0.55	0.45	0.49	0.35
Baselines fastText 300														
LSTM + Linear + CRF	0.67	0.76	0.71	0.75	0.63	0.68	0.70	0.50	0.50	0.50	0.76	0.60	0.67	0.57
Linear + CRF	0.55	0.63	0.59	0.69	0.45	0.55	0.57	0.25	0.19	0.22	0.63	0.41	0.50	0.33
LSTM + Linear + $\text{CRF}_{scratch}$	0.59	0.70	0.64	0.72	0.54	0.62	0.63	0.40	0.42	0.41	0.70	0.54	0.61	0.49
Linear + $\text{CRF}_{scratch}$	0.53	0.62	0.57	0.65	0.46	0.53	0.55	0.29	0.26	0.28	0.58	0.44	0.50	0.36
Baselines w/o embeddings														
Simple CRF	0.57	0.70	0.62	0.71	0.45	0.55	0.59	0.32	0.28	0.29	0.64	0.44	0.52	0.38
POS baseline	0.65	0.51	0.57	0.51	0.58	0.55	0.56	**0.77**	0.61	0.68	0.62	0.71	0.66	**0.67**
Majority class	0.50	**1.00**	0.74	0.00	0.00	0.00	0.43	0.60	**1.00**	**0.75**	0.00	0.00	0.00	0.45

Table 3: Average results from five different random seeds for all discourse-new vs. discourse-old sequence labeling models, probing pre-trained Transformer-XL representations vs static fastText embeddings and embeddings trained from scratch (standard deviation is between 0.01 and 0.06 for all versions). Baselines also include a simple CRF with surface features, a POS-based (pronouns and defNP = discourse-old) and the majority class (discourse-new) baseline.

combination of span + discourse-new is the most difficult category, and one in which the probes are surpassed by the baselines. Besides, the models with the LSTM yield consistently higher results than the models relying on the CRF only.

When considering the type of input, there is a similar pattern to the classifier. Although the results using static embeddings are better than the baselines, the contextualized Transformer-XL representations present a systematic improvement overall. Interestingly, this improvement is more marked for the Spans and negligible for Heads in additional experiments with a CRF for entity identification only, i.e., without labeling the entities as new or old (cf. Appendix B, Table 6). This keeps with the intuition that identifying the heads is akin to finding nouns, but identifying the relevant spans which are also entities is more complex, involving discourse-level knowledge.

4.3 Extension to other pre-trained Models

To compare our results to another pre-trained model, we also probe GPT-2 (Radford et al., 2019) in the same manner. This requires two adaptation of our approach: GPT-2 uses a different tokenizer, so the alignment of the tokenized version and the labels has to be adapted. More critically, Transformer-XL is optimized to deal with long contexts, whereas GPT-2 can only handle inputs of up to 1024 tokens. Therefore, we create a subset of our data by filtering out all documents longer than a 800 threshold (of items before tokenization). For the results to be comparable, we reran the experiments for Transformer-XL on the same subset. All the results are displayed in Table 4. We first notice that the Transformer-XL results are very similar to those obtained using the full training and test sets. Concerning the GPT-2 model, we notice that its performance is comparable to the Transformer-XL. Thus we believe that other Transformer models will be equally adequate for this task.

5 Discussion and Analysis

In line with existing literature about the presence of entity knowledge in pre-trained language models, we find that the entity knowledge extends to different types of entity mentions extracted from natural data.

The high success of the binary classifier probe demonstrates that classifying an entity as new or old is not challenging, provided that the model has access to the entity representations. Comparable to a coreference resolution system, this model is fed with an aggregated representation comprising all tokens in the mention, so it does not need to locate the entity in the sequence. On the other hand, finding the entities given a sequence is the hard part in our task, as shown by the sequence labeling probe. This model is superior in the Heads version of the data, but less successful with Spans, where the boundaries of each entity must be found. In this task, the simple POS-based baseline yields better overall results than any of our probing models.

Error Analysis: Pronouns and definite noun phrases are two types of mentions with enough reg-

	Heads							Spans						
	Discourse New			Discourse Old			Acc.	Discourse New			Discourse Old			Acc.
	Prec.	Rec.	F1	Prec.	Rec.	F1		Prec.	Rec.	F1	Prec.	Rec.	F1	
CLASSIFICATION PROBE														
Transformer-XL														
Attention-based	0.88	0.90	0.89	0.86	**0.84**	**0.85**	0.87	**0.88**	0.90	**0.89**	**0.86**	0.82	**0.84**	**0.87**
Entity-based	0.87	**0.92**	**0.90**	**0.88**	0.82	**0.85**	**0.88**	0.85	**0.91**	0.88	0.85	0.77	0.81	0.85
GPT-2														
Attention-based	**0.89**	0.90	0.89	0.86	**0.84**	**0.85**	**0.88**	**0.88**	0.88	0.88	0.82	**0.83**	0.83	0.86
Entity-based	**0.89**	0.90	**0.90**	0.86	**0.84**	**0.85**	**0.88**	0.87	0.87	0.87	0.81	0.80	0.80	0.84
SEQUENCE LABELING PROBE														
Transformer-XL														
LSTM + Linear + CRF	0.74	**0.77**	**0.75**	**0.79**	0.77	0.78	0.76	**0.55**	0.55	**0.55**	**0.78**	**0.71**	**0.74**	**0.63**
Linear + CRF	0.70	0.68	0.69	0.75	0.71	0.73	0.71	0.44	0.41	0.43	0.72	0.64	0.68	0.53
GPT-2														
LSTM + Linear + CRF	**0.76**	0.74	**0.75**	0.78	**0.81**	**0.80**	**0.77**	**0.55**	**0.56**	**0.55**	**0.78**	0.69	0.73	0.62
Linear + CRF	0.69	0.67	0.68	0.72	0.68	0.70	0.69	0.42	0.40	0.41	0.71	0.61	0.65	0.51

Table 4: Probing results **on a shortened subset of the data to accomodate GPT-2's maximum input capacity of 1024 tokens**. Results are averaged over five random seeds with a standard deviation between 0.00 and 0.04 for all versions.

ularity (i.e., closed set of forms and determiner *the*) for the models to exploit frequency heuristics. In this sense, comparing with the POS baseline constitutes an interesting case study (Table 5). In general, the fact that these two differ is a sign that our probes are not deterministically exploiting this heuristic. Note however that different genres might have different uses of definite and undefined articles, and that a definite article does not automatically entail a discourse old label.

Going into details, the first thing we observe in Table 5 is that using either the Heads or Spans version results in a similar number of errors, in particular predicting a label when there is no entity to identify (False mention). Interestingly, the sequence labeling probe yields more new than old labels (Old predicted as new), suggesting that it identifies old mentions more confidently than new ones (i.e., when it produces old, there is a high chance that the label really is old). This might explain why despite being the minority class with about 40% of the entities, discourse-old seems easier to predict in the sequence labeling experiments, in particular for the Spans setting (comparing F1 scores). In contrast, the binary classifier does slightly better with the discourse-new class. Another thing we observe is that fewer Spans are left without a prediction than Heads, which intuitively makes sense: it may be harder to say if a bare NP head is referential or not, but easier if the NP is presented with determiners, adjectives, and other modifiers. Last, the category 'Others' comprises mostly errors in detection of boundaries, which are more prevalent in the Spans (for example, *Gulf Resources & Chemical Corp. said it agreed to pay $ 1.5 million [...] regarding [an environmental cleanup] of a defunct smelter the company formerly operated*; gold: B-new I-new I-new, predicted: O O B-new). This category further suggests that finding an entity's boundaries is harder than determining its label.

Inspecting the forms closely shows that most errors correspond to the pronouns *it*, *this*, *that* and *which*, known to be problematic for coreference resolution. For the classifier, we also found that *it* and *that* are amongst the most common errors ($18/424$ in spans, $23/753$ in heads). We also inspected the definite noun phrases, but could not identify any specific pattern in the errors.

Error	Heads	Spans
False mention (gold is O)	1252	1254
No prediction (prediction is O)	138	63
New predicted as old	128	78
Old predicted as new	263	130
Others	138	456
Total	2054	1981

Table 5: Number and type of errors that the sequence labeling probe LSTM + Linear + CRF makes with respect to the POS baseline.

Model Analysis: There is no benefit in using

static embeddings (*Linear + CRF fastText*) vs simple classic features (*simple CRF*) for this task. Comparing the pre-trained representations with the models that were trained from scratch, we observe the following: i) For the most powerful model (the ones with the LSTM layer) the gain of using pre-trained embeddings is 0.15 and 0.19 in average F1 for Transformer-XL and 0.7 and 0.8 for fastText (Heads and Spans, respectively). This shows that, while the embedding size does not have an impact if embeddings are trained from scratch (similar results for 1024 and 300 dimensions), the contextualized Transformer-XL representations contain more useful information for the probing task. ii) When we look at the less powerful probing models (without the LSTM), the differences are 0.17 and 0.16 in average F1 for Transformer-XL, and 0.2 and -0.3 for fastText, showing that the LSTM is necessary to extract any useful information from the static embeddings. Transformer-XL embeddings, on the other hand, already benefit from the contextualization during pre-training, as we see similar improvements as those obtained with the complex model.

The LSTM models do yield better scores overall, suggesting that additional contextualization on the sentence level helps for this task. Collectively, these results signal that document-level contextualization does help to encode the new/old distinction, but not as much as one might have expected. If a model is presented with an entity, determining its status is not hard, even without contextualization. However, finding an entity in a sentence or discourse is challenging, even for powerful Transformer-XL representations. This raises the question of whether pre-trained language models are able to identify entities in the wild.

6 Conclusions and Future Work

In this paper, we have built two probing models for the task of identifying the discourse status of entities as new or old. Our models rely on binary classification and sequence labeling with input representations from a Transformer-XL language model. Our probes:

- have advanced the findings from previous work, showing that the discourse knowledge from pre-trained representations extends to noun phrases found in naturalistic data;
- have found that the pre-trained representations tested do encode the old/new information within the tokens comprising the entity, regardless of the context;
- have also found that localizing the entity within the sentence is difficult, suggesting that identifying referring discourse entities from scratch is hard for this pre-trained model;
- last, have demonstrated that LSTMs are able to further contextualize pre-trained static and contextualized embeddings alike.

Our findings leave interesting questions for future work, in particular, defining what an entity is and what it looks like. In this sense, one could imagine a task where a probe is asked to differentiate between referring and non-referring mentions, a known and hard problem in the context of coreference resolution.

7 Ethical Considerations and Limitations

The models trained in this study are not optimized to solve specific tasks in the best possible way, but to gain insights about the underlying representations and thus the abilities of pre-trained language models, which are sometimes attributed human like language-generation abilities. However, all findings are only applicable to the models under investigation (Transformer-XL and GPT-2) and any claims are specific to English. Reproducing our work requires access to the ARRAU corpus, on which we base all of our experiments.

Acknowledgments

SL was supported by the Swedish Research Council (VR) Grant 2014-39 for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. AB & DS were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project ID 317633480 – SFB 1287, Project B06. The authors thank Sebastiano Gigliobianco for his help extracting several versions of the data for the experiments, Yves Scherrer for his help developing the tokenizer algorithm, and Manfred Stede for valuable discussions about this work.

References

Laura Aina, Carina Silberer, Ionut-Teodor Sorodoc, Matthijs Westera, and Gemma Boleda. 2019. What

do entity-centric models learn? insights from entity linking in multi-party dialogue. 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 3772–3783, Minneapolis, Minnesota. Association for Computational Linguistics.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. Textworld: A learning environment for text-based games.

Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. 2019a. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6300–6308.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019b. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Aditya Gupta and Greg Durrett. 2019a. Effective use of transformer networks for entity tracking. 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 759–769, Hong Kong, China. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019b. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheorectic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht.

Lauri Karttunen. 1969. Discourse referents. In *International Conference on Computational Linguistics COLING 1969: Preprint No. 70*, Sånga Säby, Sweden.

Mikhail Korobov. 2015. Sklearn-crfsuite python library. Available online.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Discourse probing of pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3849–3864, Online. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of conditional random fields (CRFs). Available online.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Michael Petrochuk. 2018. Pytorch-nlp: Rapid prototyping with pytorch natural language processing (nlp) tools. https://github.com/PetrochukM/PyTorch-NLP.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 11–22, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40. Association for Computational Linguistics.

Ellen F. Prince. 1992. The ZPG letter: Subjects, definiteness, and information status. In W. Mann and S. Thompson, editors, *Discourse description: Diverse linguistic analysis of a fund-raising text*, pages 223–255. John Benjamins, Amsterdam.

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

Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5357–5367, Online. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Sebastian Schuster and Tal Linzen. 2022. When a sentence does not introduce a discourse entity, transformer-based models still sometimes refer to it. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 969–982, Seattle, United States. Association for Computational Linguistics.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for referential information in language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4177–4189, Online. Association for Computational Linguistics.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in multiple genres: the ARRAU corpus. *Natural Language Engineering*, 26:95–128.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1850–1859, Copenhagen, Denmark. Association for Computational Linguistics.

A Probing experiments details

We trained all models on an NVIDIA GeForce GTX 1080 Ti. Each classification experiment took 10 to 15 minutes, and each sequence labeling experiment took between 7 and 20 minutes.

All the classification models were trained with the BCEWithLogitsLoss and used the Adam optimizer with a learning rate of 0.001 and a batch size of 64. Early stopping was applied based on the loss on the development set.

The training for the sequence labeling probe takes between 1 and 2 epochs using early stopping based on the loss computed on the development set. We use mini-batching with size 64, a hidden size of 256 for the LSTM output, and a learning rate of 0.01 with the Adam optimizer. Dropout of 0.2 is also applied.

B Sequence labeling results without new/old labels

	Heads			Spans		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Probing Transformer-XL						
LSTM + Linear + CRF	0.86	0.91	0.89	0.76	0.76	0.76
Linear + CRF	0.84	0.79	0.81	0.61	0.59	0.60
Baselines fastText 300						
LSTM + Linear + CRF	0.86	0.89	0.87	0.69	0.68	0.69
Linear + CRF	0.75	0.75	0.75	0.45	0.35	0.39

Table 6: Single-run results from sequence labeling experiments for entity identification without predicting their status as new or old.