Probing for referential information in language models

Ionut-Teodor Sorodoc* **Kristina Gulordava** **Gemma Boleda*†**

*Universitat Pompeu Fabra
†ICREA
Barcelona, Spain
{firstname.lastname}@upf.edu

Abstract

Language models keep track of complex linguistic information about the preceding context – including, e.g., syntactic relations in a sentence. We investigate whether they also capture information beneficial for resolving pronominal anaphora in English. We analyze two state of the art models with LSTM and Transformer architectures, respectively, using probe tasks on a coreference annotated corpus.

Our hypothesis is that language models will capture grammatical properties of anaphora (such as agreement between a pronoun and its antecedent), but not semantico-referential information (the fact that pronoun and antecedent refer to the same entity). Instead, we find evidence that models capture referential aspects to some extent –though they are still much better at grammar. The Transformer outperforms the LSTM in all analyses, and exhibits in particular better semantico-referential abilities.

1 Introduction

Neural network-based language models (LMs) have been shown to learn relevant properties of language without being explicitly trained for them. In particular, recent work suggests that they are able to capture syntactic relations to a large extent (Gulordava et al., 2018; Kuncoro et al., 2018; Wilcox et al., 2018). In this paper, we extend this line of research to analyze whether they are able to capture **referential** aspects of language, focusing on anaphoric relations (pronoun-antecedent relations, as in *she-Yeping Wang* in Figure 1).

Previous work, such as Ji et al. (2017), Yang et al. (2017) and Cheng and Erk (2019), showed that augmenting language models with a component that uses an objective based on entity or coreference information improves their performance at language modeling. Intuitively, in the example in Figure 1, understanding that the first *she* refers to *Yeping Wang* makes words related to studying or working more likely to follow than other kinds of words. That is, referential information helps language models do their task.

> ... he$_1$ was elected to be president of the People's Republic of China, and chairman of the$_2$ Central$_2$ Military$_2$ Commission$_2$. Yeping$_3$ Wang$_3$ was born in Shanghai in 1926. She$_3$ studied in Shanghai Foreign Language College, and started working in 1949. For a long time, she$_3$...

Figure 1: Example from OntoNotes with a window of 60 tokens (as used in our first probe task). Both occurrences of *she* refer to the same entity as *Yeping Wang*. Note that not all entity mentions are annotated in OntoNotes –only those that enter into coreference relationships in the document.

The cited work includes explicit coreference guidance; however, since referential information is useful for language modeling, we expect language models to learn referential information even without explicit supervision. Here we analyze to what extent this is the case.

We carry out our analysis using probe tasks, or tasks that check whether certain information is encoded in a model (Adi et al., 2016; Linzen et al., 2016; Conneau et al., 2018; Giulianelli et al., 2018). The reasoning is as follows: Even if a linguistic property is encoded in the network, it is not necessarily directly accessible through the model output; therefore, we train a probe model to predict a feature of interest, in this case anaphoric coreference, given the model's hidden representations as input.

We focus on the two main linguistic levels that are relevant for coreference: **morphosyntax**, with grammatical constraints such as the fact that pronouns agree in number and gender with their an-

tecedents, and **semantics** – in particular reference, such as the fact that a pronoun refers to the same entity as its antecedent.

Our hypothesis is that language models will capture grammatical properties, but not semantic information. This hypothesis is based on the observation that morphosyntax is a formal property of language that is easier to induce from co-occurrence patterns. The fact that language refers to entities is not obvious from language alone (Harnad, 1990), and LMs use only textual input.

Instead, what we find is that, while it is true that language models are much better at grammar, they do show evidence of learning semantico-referential information to some extent. Our explanation for this unexpected, partially positive result is that, because the same entity underlies all its mentions, the contexts in which the mentions appear are coherent and distinct from those of mentions of other entities. For instance, in Figure 1, the second *she* mention gives additional information about Yeping Wang that is consistent with the information given in the previous sentence.

This paper has two main contributions. The first is an analysis methodology to probe for referential information encoded in language models, on two linguistic levels (morphosyntax, semantics) and two kinds of context: local (around one paragraph of context), and global (document context). This methodology can be applied to any architecture. The second contribution is a deeper understanding of the referential capabilities of current language models, and of the differences between Transformers and LSTMs. The Transformer outperforms the LSTM in all the analyses. For morphosyntax, the Transformer and the LSTM have the same behavior with a performance difference; instead, they show different behavior with regard to semantico-referential information.

2 Related work

Coreference and anaphora resolution (Mitkov, 2002; Poesio et al., 2016) are among the oldest topics in computational linguistics and have continued to receive a lot of attention in the last decade, as manifested by several shared tasks (Pradhan et al., 2011, 2012; Poesio et al., 2018). In our analysis we use the OntoNotes dataset (Hovy et al., 2006; Pradhan et al., 2012), developed within the coreference resolution community. Our probe tasks are related to coreference resolution; however, our goal is not to train a coreference system but to analyse whether language models extract features relevant for reference without explicit supervision.

A recent line of work has focused on demonstrating that neural networks trained on language modeling, without any linguistic annotation, learn syntactic properties and relations such as agreement or filler-gap dependencies (Linzen et al., 2016; Gulordava et al., 2018; Kuncoro et al., 2018; Wilcox et al., 2018; Futrell et al., 2018). This is typically done by analysing the predictions of LMs on controlled sets of data. Part of this research uses probe models (also known as diagnostic models) to analyse the information contained in their hidden representations (Adi et al., 2016; Conneau et al., 2018; Hupkes et al., 2018; Lakretz et al., 2019; Giulianelli et al., 2018), as we do here —applying it to referential information.

There is less work on referential information than on syntactic properties such as subject-verb agreement. As for anaphoric reference, Peters et al. (2018) include a limited test using 904 sentences from OntoNotes. Their results suggest that LMs are able to do unsupervised coreference resolution to a certain extent; our first probe task can be seen as an extended version of their task obtaining more specific insights. Jumelet et al. (2019) analyze the kind of information that LSTM-based LMs use to make decisions in within-sentence anaphora. They find a strong male bias encoded in the network's weights, while the information in the input word embeddings only plays a role in the case of feminine pronouns. We analyze anaphora in longer spans (60 tokens / whole document) and include also a Transformer.

The above work suggests that LMs capture morphosyntactic facts about anaphora to a large extent. There is much less evidence that LMs can capture a notion of entity, as that which nominal elements refer to, and that they are able to track entities across a discourse. Parvez et al. (2018) show that LSTM-based models have poor results on texts with a high presence of entities; Paperno (2014) that they cannot predict the last word of text fragments that require a context of a whole passage (as opposed to the last sentence only), with data that mostly contain nominal elements. Several models (Henaff et al., 2019; Yang et al., 2017; Ji et al., 2017) were developed as an augmentation of RNN LMs to deal better with entities, with the

implicit assumption that standard models do that poorly. Aina et al. (2019) achieved good results on an entity-linking task, but showed that the network was not acquiring entity representations.

As for Transformer-based architectures, recent research suggests that they give same or better contextualized representations in comparison with LSTM language models, and that they better encapsulate syntactic information (Goldberg, 2019; Wolf, 2019). On the other hand, van Schijndel et al. (2019) show that big Transformer model representations perform on par or even poorer than smaller LSTMs on tasks such as number agreement or coordination, and that, like LSTMs, they have the problem that agreement accuracy decreases as the subject becomes more distant from its verb. Most recent work on analysis of linguistic phenomena in NNs focuses on BERT (Tenney et al., 2019; Clark et al., 2019; Reif et al., 2019; Broscheit, 2019). In this paper we chose to use TransformerXL (Dai et al., 2019) as our Transformer model, and not BERT, for comparability: We wanted to compare the two most standard architectures for LMs on as equal ground as possible, and the two chosen models, TransformerXL and AWD-LSTM (Merity et al., 2017), share the same training objective and are trained on the same data, with comparable vocabularies.

3 Morphosyntactic factors

To shed light into which morphosyntactic information LMs encode that is useful for coreference, we train a simple anaphora resolution probe model using the hidden layers of LMs as input. By the logic of probe tasks, if the probe model is successful then that means that the relevant information is encoded in the hidden states, and error analysis can provide insight into which kinds of information are available.

3.1 Experimental Setup

Data We train our probe models on data from OntoNotes 5.0 (Weischedel et al., 2013). We use the annotated coreference chains, as well as the provided part-of-speech tags (the latter only for analysis purposes).

We take all pronouns that have at least one antecedent in a 60-token context window; the task of the probe model is to identify their antecedent.[1]

[1]We also experimented with windows 20 and 200, obtaining a similar picture.

	Tokens	Datapoints
Train	191,830	4,949
Dev	275,201	4,556
Test	2,026,565	45,665

Table 1: Dataset statistics for first probe task. We reverse the original train and test partitions (see text).

An example datapoint is provided in Figure 1 above (note that a window of 60 tokens allows us to check anaphora beyond the sentence). For simplicity, antecedents are tokens, but typically there is more than one possible token antecedent for a given pronoun: A mention can span several tokens (*Yeping Wang*), and the window can contain several mentions from the same coreference chain (*Yeping Wang* and the first *She* in Figure 1); we consider any of the tokens a correct answer. Note that we are not training the model to explicitly identify mentions, their spans or the complete coreference chains, but to identify the tokens that are antecedents of the target pronoun.

To obtain enough data for analysis, especially for low-frequency phenomena, we follow Linzen et al. (2016) in reversing the original partitions of the corpus, using the original test set for training and the original training set for testing.[2] In addition, we focus on the OntoNotes documents that belong to narrative text sections because the dialogue data does not come with turn segmentation.[3] Resulting data statistics for our task are provided in Table 1.

Language models The base language models we use are AWD-LSTM (Merity et al., 2017) and TransformerXL (Dai et al., 2019), two state-of-the art models with the most standard architectures for language modeling as of 2020 (LSTM, Transformer). We chose these models for compar-

[2]Using little training data has also been shown to lessen the possibility of confounds in the probe model results; in particular, it makes it more difficult for the probe model to exploit regularities in the training data rather than capturing the analyzed model's ability to capture a phenomenon (Hewitt and Liang, 2019). See Voita and Titov (2020) for a theoretical justification from a information-theoretic perspective.

Results on the original split confirm that the conclusions of the paper are robust: we see an increase in performance of around 3% overall, as could be expected because we use more data, but the same behavior patterns (on the data that can be compared).

[3]We keep newswire (NW), broadcast news (BN), magazine (MZ), web data (WB), and pivot text (PT), removing broadcast conversation (BC), telephone conversation (TC).

ison because they are trained on the same dataset (Wiki103; Merity et al., 2016), they have a comparable vocabulary, and they are both very strong language models, with perplexities of 24 for TransformerXL and 33 for AWD-LSTM. TransformerXL is a bit larger than AWD-LSTM, though (151 million parameters compared to 126), which should be kept in mind when assessing results.[4]

Probe model For each word x_i in the window of size m preceding the target pronoun x_t, we obtain its contextualized representation h_i from the last hidden layer of the language model (Eq. 1). The probe model takes this representation as input and is trained to map it onto a vector o_i using a non-linear transformation (Eq. 2). The target pronoun representation is transformed in the same way. The dot products between these transformed representations of target and context word vectors give the attention weights ref_i (Eq. 3) representing the similarity between two representations. The weights are transformed into probabilities using the softmax function (Eq. 4). Like this we obtain a probability distribution p_i over context tokens.

During training, the probe model's objective is to assign higher probabilities (and thus attention weights) to correct antecedents, and lower probabilities to incorrect ones, through the use of the Kullback-Leibler divergence loss (Eq. 5). We use the KL loss because we frame the task in terms of a probability distribution over mentions in the context. For the reasons discussed above, there can be $k > 1$ correct predictions out of m tokens in the window. We assume that gold probability distribution is uniform over k correct tokens, that is, each of these tokens has a probability $p_i^* = \frac{1}{k}$ and all other tokens have a probability of 0.[5]

[4]We also trained an in-house LSTM on data that are more similar to those of OntoNotes and a smaller vocabulary. The results for this model (not reported) follow the same patterns as those found for the AWD-LSTM and TransformerXL models, although the performance on this probe task is much higher than that of AWD-LSTM.

[5]Note however that minimizing KL divergence and minimizing cross-entropy gives the same results, because $KLdiv(p||q) = CrossEntropy(p,q) - entropy(p)$, and $entropy(p)$ is constant. Technically, in PyTorch the cross-entropy loss is only implemented for classification task targets, while the more general KL loss is available for predicting probability distributions.

Model	Accuracy	
closest gold entity	56.1	
closest same-form token	61.3	
	unsup.	sup.
LSTM	41.7	64.8
Transformer	48.5	**75.9**

Table 2: Probe model results on anaphora resolution.

$$h_i = LSTM(x_i) \quad (1)$$
$$o_i = ReLU(W * h_i + b) \quad (2)$$
$$ref_i = o_i \odot o_t, \forall i \in [t-m, t-1] \quad (3)$$
$$p_i = softmax(ref_i), \forall i \in [t-m, t-1] \quad (4)$$
$$L = KL(p_i, p_i^*) \quad (5)$$

As mentioned above, we fix $m = 60$. We train the probe model for 50 epochs with a learning rate of 1e-5 and ADAM as optimizer. The transformed vectors o_i have a dimensionality of 650 in the case of both models in comparison with h_i which is 400 for the AWD-LSTM and 1024 for TransformerXL.

Baselines We report two rule-based baselines that give relatively good performance in anaphora resolution: Referring to the previous entity (given by the oracle gold annotation; in Figure 1, *she* would refer to the previous *She*), and always pointing to the token in the window that has the same form as the target pronoun (that is, in Figure 1, *she* → *She* —we ignore capitalization). In addition, to compare the result of the probe model with the input representations, we also report an unsupervised baseline: Referring to the token in the window that has the highest similarity $cos(h_i, h_t)$ to the target pronoun, i.e., relying on the similarity between the non-transformed hidden representations.

3.2 Results

Table 2 summarizes the results of the pronominal anaphora probe task. The probe model trained on top of the LSTM improves a bit over the strongest baseline, and that of the Transformer does so substantially (75.9 vs. 61.3; the LSTM obtains 64.8). This performance suggests that the LMs use more information than simple heuristics like referring to a token with the same form.

The unsupervised similarity baseline performs worse than the rule-based baselines. This is to

be expected: The "raw" similarity between hidden states is based on many more aspects than those related to reference, given that hidden states are responsible for capturing all the contextual features that are relevant for word prediction. This is why a probe model is needed to distill the reference-related information from the hidden layers.

A single non-linear layer trained on only 5K datapoints improves performance by 23-28 absolute accuracy points (supervised vs. unsupervised results), which suggests that the referential information in the hidden layers is easy to extract. Behaviorally, the unsupervised hidden layers are quite similar to the baselines. First, they are biased towards tokens of the same form: in 27.1% of the cases, the LSTM layer of the pronoun presents the highest similarity to a token with the same form; 29.1% in the case of the Transformer. Second, they prefer close antecedents, although the LSTM presents this recency bias to a much higher degree: in 27.8% of the cases, the LSTM layer of the pronoun has the highest similarity to the previous token (16.4% in the Transformer). The attention mechanism of the Transformer gives access to a broader context and allows it to overcome the recency bias to some degree.

The great difference in performance between AWD-LSTM and TransformerXL could suggest that the latter is using different strategies compared to the former. Instead, except for the recency bias, what we find are exactly the same patterns in behavior, with a systematic 10% accuracy gap. For this reason, although we provide results for both models everywhere to show that this observation indeed holds, in this section we will mostly focus on the Transformer when commenting results.

3.3 Analysis: Morphosyntactic Factors

The models clearly learn grammatical constraints related to anaphora that are well-studied in the literature and are relied upon by traditional anaphora resolution models (Sukthanker et al., 2018). First, as shown in Table 3, the Transformer identifies mentions (elements inside some coreference chain) in 92.6% of the cases. Moreover, it correctly learns that pronouns typically refer to nominal elements (almost 95% identified antecedents are pronouns, proper nouns, and elements within a noun phrase headed by a common noun). Note that pronouns can also have non-nominal antecedents, although these are the minority of the annotations in OntoNotes (cf. example 4 in Figure 3, where *it* refers to an event). Even in the cases in which the Transformer points to elements outside of a chain (7.4%), it points to nominal elements 87% of the time (not shown in the table). The model is most accurate when referring to pronouns (82.6% accuracy), while noun phrases are the hardest category (62.3%). This is consistent with the strategies that the model learns, since it largely relies on pronominal agreement, as described below.

	LSTM 90.2% in chain		Transformer 92.6% in chain	
POS	Perc.	Acc	Perc.	Acc
Noun phrase	15.5	50.9	17.0	62.3
Proper noun	20.2	64.3	20.0	74.9
Pronoun	59.0	71.5	59.0	**82.6**
Other	5.3	67.3	3.0	81.6

Table 3: Statistics on types of mentions that the probe models refer to, for predictions that are in a coreference chain. 'Noun phrase' stands for elements that are typically within a noun phrase (note that our system points to individual tokens): Determiners, nouns, and adjectives.

Second, not only do the models mostly point to nominal elements, but they also identify the morphosyntactic properties of pronouns and learn that they should agree with their antecedents in gender and number. Figure 2 shows the distribution of pronoun antecedents that the Transformer predicts, for the six most frequent target pronouns (see the Supplementary material for the corresponding LSTM figure). Its preferred type of antecedent are pronouns of the same form, but it is also able to point to other pronouns agreeing in number and gender. For instance, pronoun *he* points to 3rd person, masculine, singular pronouns (mostly *he*, but also *his*, *him*) —a pattern consistent across all pronouns.

Figure 2 is restricted to pronouns; Table 4 shows that the model also largely follows number agreement when predicting antecedents within noun phrases (the table collapses common noun and proper noun antecedents). Given a singular pronoun, the model chooses a singular antecedent 98% of the time; given a plural pronoun, it identifies a plural antecedent in 73% of the cases.

Note that in cases of plural pronouns such as

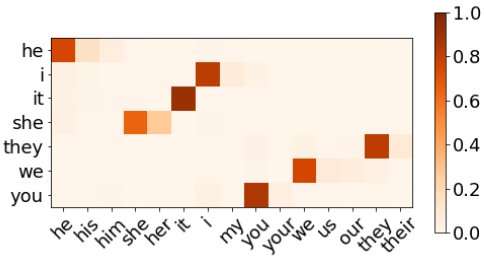

Figure 2: Pronominal agreement with Transformer probe model: Proportion of cases in which elements in the rows corefer with elements in the columns.

	LSTM		Transformer	
Pron-ant.	Perc.	Acc	Perc.	Acc
sg-sg	97.7	66.3	98.7	**76.0**
sg-pl	2.3	20.5	1.3	36.7
pl-sg	35.5	40.8	27.5	53.1
pl-pl	64.5	67.7	72.5	**72.3**

Table 4: The types of noun phrase antecedents the models choose, by number agreement (e.g., 'sg-pl' means 'anaphoric pronoun is singular, antecedent plural').

they it is common that the referent be a singular noun (e.g., *the audience* in example 3, Figure 3), reflected by the reasonable accuracy of the Transformer in pl-sg cases (53.1%).

4 Semantic (referential) factors

The language model clearly captures morphosyntactic (grammatical) properties that constrain anaphora resolution; in this section, we show that it struggles more with is the semantic (referential) aspect, but it still captures it to some extent.

4.1 Sensitivity to distractors

If the model were able to model entities, it should be robust to *distractors*, that is, mentions in the context that are not antecedents –in Figure 1, *he* and *the Central Military Commission*. Figure 4 shows that the accuracy for the Transformer decreases as does the proportion of gold mentions. We compute this proportion as the number of gold mentions in the 60-token window divided by the total number of mentions in the same window. When there are no distractors (gold proportion = 1), accuracy is very high, which is to be expected given that the model learnt to identify mentions in the first place (cf. previous section). The more distractors (i.e., the lower the proportion of gold mentions), the lower the accuracy; however, accuracy decreases rather gracefully. Even when there are only 10% gold mentions in the window, accuracy for most pronoun types is still around 60-80%. The exception is *it*, which is the most difficult pronoun for the model, presumably because it can refer to many kinds of antecedents.[6]

Figure 4 thus paints a nuanced picture: distractors confuse the model, but they do not fool it completely. Given the results in the previous section, we expect that distractors sharing morphosyntactic features will be particularly challenging. Table 5 confirms this, zooming in into pronominal distractors. We consider a datapoint having a pronominal distractor if one of the antecedents is a pronoun pointing to another entity.

When there are no pronominal distractors (25.9% of the test set), the accuracy of the Transformer is 81.8%; with at least one distractor, it goes down to 73.8% —clearly worse but not dramatically so. However, in cases where anaphoric pronoun and antecedent have the same gender, number, or are the same pronoun, we get much lower accuracies (48.6, 65.3, and 49.1, respectively). This suggests that that the model overly relies on morphosyntactic features and recency (see previous section).[7]

However, accuracy in these cases goes down but is still decent, compared to a reasonable baseline (last column in the table). For each target anaphoric pronoun, we calculate baseline accuracy as the percentage of gold pronouns in the window (pronouns that are in the same chain as the target), that is, number of gold pronouns divided

[6] While most personal pronouns refer to people, which are relatively homogeneous kinds of referents, *it* refers to very varied kinds of referents. Qualitative analysis suggests that the model is quite successful when *it* refers to concrete entities (*province*, *peanut*), but much less when it refers to abstract objects like propositions or events, as in example 4 of Figure 3 (where *it* refers to the event of trying to improperly influence a witness). A quantitative check confirms this hypothesis: Cases in which the model fails have around 18% of verbal references, compared to less than 2% for cases in which the model is right.

[7] Among the hardest cases are those where two coreference chains in the window have the same pronoun (e.g. *he*) or gender (e.g. *he-his*). Most of these cases appear when the text includes reported speech (see Figure 3, example 1). Otherwise, there are few cases of such local ambiguity, which is presumably avoided by language speakers. However, qualitative analysis suggests that the presence of distractors is also problematic in the case of nouns, as illustrated in example 2 of Figure 3, where the model is presumably confused by a noun of the same gender and number as the pronoun (*priest* vs. *Peter-him*).

1. Why had **Mr. Korotich** been called? "I told my driver," *he* said, "that **he**
2. While **Peter** was still in the yard, a servant girl of the high *priest* came there. She saw him warming himself by the fire. She looked closely at **him**
3. The performance by more than 40 members of the Rome Philharmonic Orchestra intoxicated ***the audience*** and the musical fountain, hi-fi sound effect, fountain screen and stereographic projection brough **them**
4. Mr. Gonzalez expressed concern over *a report* that the two had been summoned to Washington by Mr. Wall last week to discuss their testimony in advance. "I think he is **trying** to improperly influence a witness, and by God I 'm not going to tolerate **it**

Figure 3: Difficult cases of anaphora. The target pronoun and its antecedent are in **bold**; the prediction of the model is in *italic*.

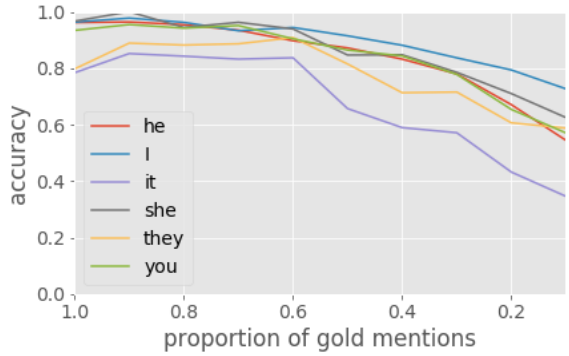

Figure 4: Transformer probe model: Accuracy as a function of the proportion of mentions that are antecedents (vs. distractors) in the window.

Type	Perc.	L Acc.	T Acc.	Base Acc.
No distractor	25.9	74.9	**81.8**	100
Distractor(s)	74.1	61.3	73.8	32.0
= gender*	4.8	40.9	48.6	15.7
= number	37.2	55.7	65.3	26.6
= pron.	10.3	39.7	49.1	20.3

Table 5: Percentage of datapoints with/without pronominal distractors and accuracy of the models (LSTM - L, Transformer - T) and baseline (last column). *Excludes cases with no marked gender (like *I*, *you*).

by the total number of pronouns in the window. Then we calculate the average of this accuracy over the respective subset (no distractors / distractors / same gender, etc.). The baseline when there are no distractors is by definition 100%; when there are distractors, it ranges between 15.7 and 32%. All model accuracies are well above this baseline.

The results thus suggest that the models are able to distinguish mentions of different entities to some extent, although they are far worse at this than at capturing morphosyntactic features. In the following subsection, we provide further support for this interpretation.

4.2 Distinguishing entities

Our last piece of analysis looks at whole documents. We aim at testing whether the hidden representations of the language models contain information that can help distinguish mentions of the same entity from mentions of some other entity, even if they are of the same form; for instance, a pronoun *she* referring to two different women. We use coreference chains to identify the tokens referring to the same entity, and train a probe model to determine when two pronouns are referring to the same entity, that is, whether they are part of the same coreference chain in a document. In the previous probe task, where the model was trained to find a correct local antecedent, the model could use cues such as linear distance and syntactic relations; here it should rely on more persistent entity-related features in the hidden representations.

Experimental Setup. We focus on pronouns because they cannot be disambiguated on the basis of lexical features. We use the same train/test partition as in the first probe task. For each datapoint, we have two pronouns: **x** and **y**, which can either come from the same chain, or not. Again, we take each pronoun to be represented by the last hidden layer representation of the language model (Eq. (1)): h_x and h_y. We call this representation *un-*

supervised, and will compare it to the supervised one, obtained as follows.

Similarly to the previous probe task, the embeddings are transformed through a learnt linear transformation to a 400-dimensional vector to extract features relevant for the entity identification task (Eqs. (6) and (7)). We take the cosine between the transformed representations as the similarity between the two pronouns.

We take as positive datapoints contain two pronouns belonging to the same chain, as negative datapoints two pronouns from two different chains. During training, for each document, we extract all positive pairs and then randomly select the same number of negative pairs. The model optimises max-margin loss on these datapoints (Eq. (8), where x and y belong to the same chain and x' and y' belong to two different chains).

$$o_x = W * h_x + b \tag{6}$$

$$o_y = W * h_y + b \tag{7}$$

$$L = 1 - cos(o_x, o_y) + cos(o_{x'}, o_{y'}) \tag{8}$$

Results Figure 5 plots the similarities between positive and negative pairs (solid and dashed lines, respectively) for the two analyzed language models, compared to linear distance in the text. The left graph corresponds to unsupervised similarities, the right graph to supervised similarities. To control for token form effect, we only include data with the same pronoun pairs in this graph. Three results stand out. First, despite training with a global objective, with no linear information, similarities are negatively correlated with linear distance in text. This is consistent with the tendency of the unsupervised cosine baseline of pointing to the closest token (see Section 3).

The second result is that, crucially, after controlling both for distance and for pronoun form, similarities are systematically higher for coreferring pronoun pairs than for non-coreferring ones. Thus, some properties make their way into the hidden representations (and the probe model) that make coreferring mentions distinct from non-correferring mentions —modulo distance: If we attempt to globally distinguish chains, we instead obtain null results (see Supplementary Materials). This is because, with linear distance, the similarity in the entity-centered representation space shrinks very fast; same-chain mentions that are further away have lower average similarities than different-chain mentions that are nearby.

Finally, the third main result is that the supervised model is able to extract discriminating information from the hidden layers to a much larger extent in the Transformer than in the LSTM (cf. distance between blue and red lines, respectively). We interpret this to mean that such information is encoded to a larger extent in the Transformer. Also note that the supervised LSTM model is more sensitive to linear distance than any of the other representations (cf. the steeper curves between 0-100 token distances). As we signaled in the previous section, LSTM is more prone to recency biases, and it looks like global representations contain less entity-related information than in the case of the Transformer, such that the supervised model defaults to recency. We conclude from this that the Transformer accounts for semantico-referential aspects better than the LSTM.

Overall, the results suggest that token form and proximity in text remain the main properties encoded in the hidden states of entity mentions, but other properties that discriminate between coreferring and non-corefering mentions are present to some extent, allowing for partial discrimination.

5 Conclusion

Previous work has provided robust evidence that language models capture grammatical information without being explicitly trained to do so (Linzen et al., 2016; Gulordava et al., 2018). In this paper, we have analyzed to what extent they learn referential aspects of language, focusing on anaphora. We have tested two models representative of the prevailing architectures (Transformer, LSTM), and our methodology can be extended to any other architecture.

We find that the two models behave similarly, but the Transformer performs consistently better (around 10% higher accuracy in the probe tasks).[8] Future work should test other architectures, like CNN-based LMs and LSTMs with attention, to provide additional insights into the linguistic capabilities of language models.

As expected, our results show that language models capture morphosyntactic facts about anaphora: Based on the information in the hidden layers, a simple linear transformation learns to link

[8] With the caveat that the model we tested is slightly bigger than its LSTM counterpart.

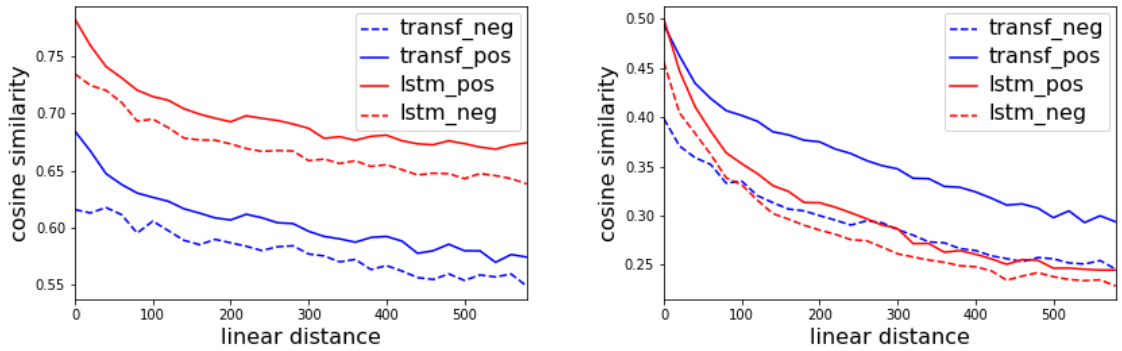

Figure 5: Linear distance in the discourse vs. cosine distance, for all the mention pairs with the same token pronoun. Distances averaged within bins of 20 tokens. Left: unsupervised, right: supervised.

pronouns to other pronouns or noun phrases, and to do so largely respecting agreement constraints in gender and number.

Although it is much harder for models to induce a more global notion of entity (what we have called semantico-referential aspects), models seem to encode entity-specific information to some extent. Models get confused when there are other mentions in the context, especially if they match in some morphosyntactic feature, but less than could be expected; and they show some limited ability to distinguish mentions that have the same form but are in different coreference chains, though hampered by their heavy recency bias. The recency bias affects LSTMs more, but is also found in Transformers, consistent with previous work on syntax (van Schijndel et al., 2019).

Our results thus suggest that language models are more successful at learning grammatical constraints than they are at learning truly referential information, in the sense of capturing the fact that we use language to refer to entities in the world; however, they still do surprisingly well at referential aspects, given that they are trained on text alone. Future work should investigate where these primitive referential abilities stem from and how they can be fostered in future architectures and training setups for language modeling, and neural models more generally.

Acknowledgments

We gratefully acknowledge the AMORE team for the feedback, advice and support. We are also grateful to the anonymous reviewers for their valuable comments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 715154), and from the Spanish Ramón y Cajal programme (grant RYC-2015-18907). We thankfully acknowledge the computer resources at CTE-POWER and the technical support provided by Barcelona Supercomputing Center (RES-IM-2019-3-0006). We are grateful to the NVIDIA Corporation for the donation of GPUs used for this research. We are also very grateful to the Pytorch developers. This paper reflects the authors' view only, and the EU is not responsible for any use that may be made of the information it contains.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Laura Aina, Carina Silberer, Ionut 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.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685.

Pengxiang Cheng and Katrin Erk. 2019. Attending to entities for better text understanding. *arXiv preprint arXiv:1911.04361*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation Without Parallel Data. In *ICLR 2018*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. 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.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2019. Tracking the world state with recurrent entity networks. In *5th International Conference on Learning Representations, ICLR 2017*.

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.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

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.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11.

Leonard Kaufman and Peter J Rousseeuw. 1990. *Finding groups in data: an introduction to cluster analysis*. John Wiley, New York.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Yair Lakretz, Germán Kruszewski, Théo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

R. Mitkov. 2002. *Anaphora Resolution*. Studies in Language and Linguistics. Longman.

Denis Paperno. 2014. Typology of adjectives benchmark for compositional distributional models. In *LREC*.

Md Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2373–2383.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

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.

Massimo Poesio, Roland Stuckardt, and Yannick (Eds.) Versley. 2016. *Anaphora resolution: Algorithms, resources, and applications*. Springer.

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, Jeju Island, Korea. Association for Computational Linguistics.

Sameer Pradhan, Lance A. Ramshaw, Mitchell P. Marcus, Martha Palmer, Ralph M. Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*, pages 1–27.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. 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 5835–5841.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2018. Anaphora and coreference resolution: A review. *arXiv preprint arXiv:1805.11824*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf. 2019. Some additional experiments extending the tech report "assessing BERT's syntactic abilities" by Yoav Goldberg. Technical report.

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.

A Additional results for first probe task (local context)

The probe models tend to refer to entities that are further away from the target than the closest *gold* entity (74.2% cases in the case of the Transformer), suggesting that they do not rely on a simple recency bias either (although both models do exhibit a recency bias, as we show in the main paper). This observation is confirmed when looking at the distribution of predicted antecedents and gold antecedents (Figures 6 and 7).

Figure 8 presents a heatmap of pronominal agreement for AWD-LSTM. Similar to the TransformerXL heatmap from the main paper, we can observe that in the majority of cases, the model predicts same form tokens with some variation either at the gender level or at the number level.

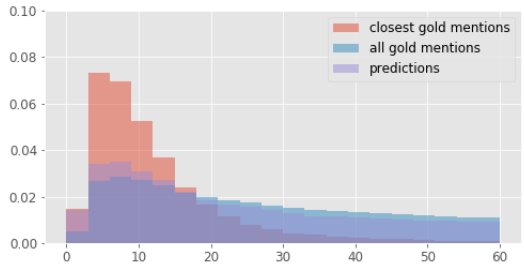

Figure 6: The distances between the pronoun and its gold and predicted antecedents for TransformerXL.

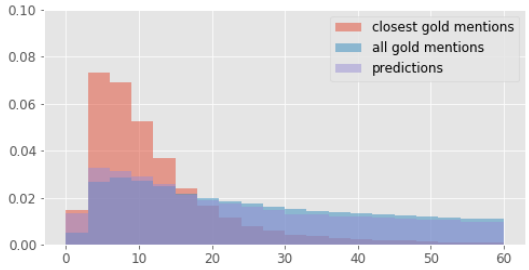

Figure 7: The distances between the pronoun and its gold and predicted antecedents for AWD-LSTM.

Figure 9 presents the performance of AWD-LSTM relative to the number of distractors in the window. While the tendencies seem to be the same as the ones for TransformerXL, the curves are steeper, the model being more confused with a higher number of distractors.

B Additional results for second probe task (global context)

In the main text, we say that, if we attempt to globally distinguish chains, we obtain null results. Here we show the results of the experiment that leads to these null results.

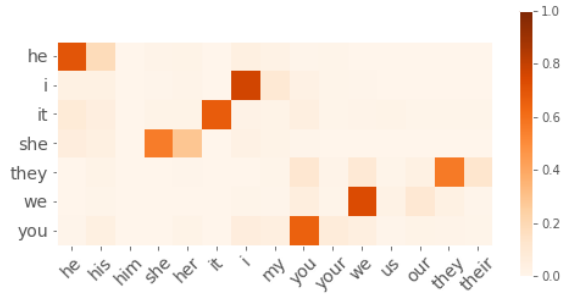

Figure 8: Pronominal agreement: Proportion of cases in which elements in the rows refer to elements in the columns for AWD-LSTM

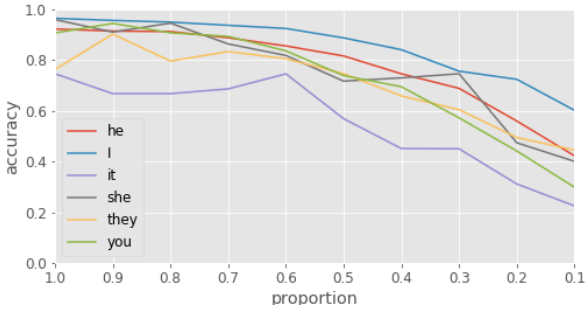

Figure 9: The accuracy of reference with respect to the ratio of correct versus confounding mentions in the window for AWD-LSTM

Method To evaluate the distance metric learnt by the model we use the silhouette coefficient (Rousseeuw, 1987), which is commonly used for intrinsic clustering evaluation. The silhouette coefficient for each pronoun **x** is defined as in Eq. (9), where a is the mean distance between **x** and all other items in the same chain, and b is the mean distance between **x** and all other items in the closest chain (measured in the learnt space, not in terms of linear distance). Its range is $[-1, 1]$, with 1 corresponding to the pronoun being much closer to the other pronouns in its chain, 0 being borderline (equally close to the two compared chains), and -1 being much closer to the pronouns in the other chain. The average silhouette coefficient is used as an overall measure of clustering quality. A score below 0.25 is usually deemed a null result (Kaufman and Rousseeuw, 1990).

$$s = \frac{b - a}{max(a, b)} \tag{9}$$

The probe model is trained for 50 epochs, keeping the model at the best validation epoch, i.e., where the silhouette score over the validation data is highest.

In addition to the trained probe model, we provide the results on global entity discrimination for the unsupervised baseline which computes the cosine similarity between the non-transformed hidden representations of the language models, similarly to the first probe task.

Results and Discussion All the obtained values are well below 0.25. Table 6 contains the results for all the datapoints as well as divided into easy and difficult documents. In easy documents, all the chains have different pronouns, so they can be distinguished by the token form only. Difficult documents contain confusable chains, that is,

there are at least two different chains which share the same pronoun. Coefficients are a bit higher for easy documents, but still very low, and, for complex documents, they are virtually zero. Moreover, the supervised models performs marginally better than the cosine baselines, but clearly do not learn any reliable information.

		LSTM		Transformer	
	N	unsup	sup	unsup	sup
all	1142	-0.09	0.02	-0.08	0.03
easy	194	0.12	0.14	0.13	0.16
diff	948	-0.13	-0.007	-0.13	0.01

Table 6: Results for the second probe task (average silhouette coefficient).

Indeed, the average distances within and across chains seem to confirm these results. If models were capturing global entity-related properties in their mention representations, we would expect pronouns with the same form but in different chains to be further away than pronouns (of any form) that belong to the same chain; instead, they are at the same distance (average cosines of 0.75 / 0.76 for Transformer, 0.74 / 0.73 for LSTM, respectively).

We conclude that the models' sensitivity to whether two identical pronouns belong to the same chain or not only shows if linear distance is factored out (as in the main text). If it is not, as in the current experiment, the models fail completely at distinguishing entities.