Sentence Ambiguity, Grammaticality and Complexity Probes

Sunit Bhattacharya and Vilém Zouhar and Ondřej Bojar
Charles University, Faculty Of Mathematics and Physics
Institute of Formal and Applied Linguistics
{bhattacharya,zouhar,bojar}@ufal.mff.cuni.cz

Abstract

It is unclear whether, how and where large pre-trained language models capture subtle linguistic traits like ambiguity, grammaticality and sentence complexity. We present results of automatic classification of these traits and compare their viability and patterns across representation types. We demonstrate that template-based datasets with surface-level artifacts should not be used for probing, careful comparisons with baselines should be done and that t-SNE plots should not be used to determine the presence of a feature among dense vectors representations. We also show how features might be highly localized in the layers for these models and get lost in the upper layers.

1 Introduction

Pre-trained language models, such as BERT, M-BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), while being very efficient at solving NLP problems, are also notoriously difficult to interpret and their analysis and interpretation is an active area of research (Belinkov and Glass, 2019). One such technique of analysis is based on probing classifiers (Belinkov, 2021), which primarily consists of training and evaluating a shallow network multi-layer perceptron (MLP) as a classifier on top of the vector representations. Probing classifiers are now fairly established in NLP (Adi et al., 2016; Tenney et al., 2019; Ma et al., 2019).

In this work, we build sentence representations from layer-wise contextual embeddings obtained from three different pre-trained language models and probe them for three linguistic traits: sentence ambiguity, grammaticality, and complexity using some well-established datasets.

In the process, we show why having a reasonable baseline is a necessity for performance interpretation. We also demonstrate why simply visually checking the clustering of embeddings on datasets using t-SNE, a popular dimension-reduction technique in probing, can lead to incorrect conclusions.

Motivation. The study of these traits is important for example in machine translation where disambiguation is necessary and grammaticality correction and simplification sometimes happen implicitly without any control. For the tasks of text simplification and grammar correction, it is crucial to be aware of whether and how general-purpose models encode these traits or whether they abstract the meaning from them. Specifically, ambiguity detection has been investigated very little in contrast to other features. All of these three traits are orthogonal in their definitions, although their mutual relationships are unknown. For example, it may be that ambiguous sentences tend to be more complex and prone to lower grammaticality. We assimilate the definition of these traits from the respective datasets but nevertheless include examples in Table 1.

Contribution. We carry out text classification tasks of ambiguity, grammaticality and complexity and demonstrate empirically that:

• having a reasonable baseline is a necessity for performance interpretation;
• sentence ambiguity is represented much less than sentence complexity in the models;
• the template-based BLiMP dataset is not suitable for probing grammaticality because of surface-level artefacts;
• t-SNE is not always an adequate tool to see whether a feature is represented in vectors.
### 2 Related Work

#### Ambiguity.
Word-sense disambiguation has been extensively studied and is a closely related task (Navigli, 2009). This has also been the focus of work done with recent NLP tools, which has mostly concentrated on the determination of ambiguity at the lexical level and not at the sentence level. Yaghoobzadeh et al. (2019); Şahin et al. (2020); Meyer and Lewis (2020) classify ambiguous words. Chen et al. (2020) explore the geometry of BERT and ELMo (Peters et al., 2018) using a structural probe to study the representational geometry of ambiguous sentences. Bordes et al. (2019) use a combination of visual and text data to ground the textual representations and make notes on disambiguation. Ambiguity modelling has also been a focus of the MT community because translation often requires disambiguation. This applies on many levels: lexical (HiginoBotham, 1991; Zou and Zou, 2017; Do et al., 2020; Campolungo et al., 2022), syntactic (Pericliev, 1984) and semantic (Baker et al., 1994; Stahlberg and Kumar, 2022). Psycholinguists have also studied the effect of ambiguity resolution on cognitive load (Altmann, 1985; Trueswell, 1996; Papadopoulou, 2005), often motivated by issues in MT (Sammer et al., 2006; Scott, 2018). Bhattacharya et al. (2022) explore ambiguity by the task of translation by human annotators.

#### Grammaticality.
This trait has been studied historically from the perspective of human sentence processing and acceptability (Nagata, 1992; Braze, 2002; Mirault and Grainger, 2020). Many real-world applications utilize tools for automatic grammaticality prediction (Heilman et al., 2014; Warstadt et al., 2019), such as automatic essay assessment (Foltz et al., 1999; Landauer, 2003; Dong et al., 2017) or machine translation (Riezler and Maxwell III, 2006). For MT, output acceptability, or fluency, is a standard evaluation direction for which many automated metrics exist (Hamon and Rajman, 2006; Lavie and Denkowski, 2009; Stymne and Ahrenberg, 2010). In contrast to our supervised classifier approach, perplexity-based approach has been used to measure acceptability (Meister et al., 2021).

Related more closely to our setup, Hewitt and Manning (2019) use a linear probe and identify syntax in contextual embeddings. Lu et al. (2020); Li et al. (2021) examine gramaticality in BERT layers. Hanna and Bojar (2021) assess BERTScore effectiveness in spotting grammatical errors.

#### Complexity.
Similarly to other traits, complexity was first studied in the human processing of language (Richek, 1976; Just et al., 1996; Heinz and Idsardi, 2011). Brunato et al. (2018) perform a crowd-sourcing campaign for English along with an in-depth analysis of the annotator agreement and complexity perception. Automatic complexity estimation is vital, especially in the educational setting for predicting readability (McNamara et al., 2002; Weller et al., 2020). Ambati et al. (2016) estimate sentence complexity using a parser while...
Štajner et al. (2017) do so using n-grams. Sarti (2020); Sarti et al. (2021) juxtapose the effect of complexity on language models and human assessment thereof. Martinc et al. (2021) survey multiple neural approaches to complexity estimation, including using pre-trained LM representation. In contrast to our work, they report only the final results and do not investigate the issue from the perspective of probing (e.g. what representation to extract and from which layer).

**Probing.** Earlier probing studies have shown that the early layers of BERT capture phrase-level information and the later layers tend to capture long-distance dependencies (Jawahar et al., 2019). The syntax is also captured more in the early layers of BERT and higher layers are better at representing semantic information (Tenny et al., 2019). It is not clear if and how pre-trained models achieve compositionality (Kalchbrenner and Blunsom, 2013; Nefdt, 2020; Kassner et al., 2020) and how linguistic knowledge is represented in sentence embeddings. Liu et al. (2019) use probing on a set of tasks including token labelling, segmentation and pairwise relation extraction to test the abilities of contextual embeddings. Mutual information can be used as a viable alternative to traditional probes that require optimization (Pimentel et al., 2020). A conceptual follow-up is *V*-information (Hewitt et al., 2021) which is better suited for probing. In many cases, t-SNE is the prevalent method of visualization of class clusters in high-dimensional vector space (Jawahar et al., 2019; Jin et al., 2019; Wu and Xiong, 2020; Hoyt and Owen, 2021).

### 3 Data

For each trait, we use a different dataset. Their overall sizes are listed in Table 2 and example sentences in Table 1. We repurpose the datasets and derive binary labels (positive/negative) from each: ambiguous/unambiguous, complex/simple and grammatical/ungrammatical.

**Ambiguity.** We use sentences from the MS COCO (Lin et al., 2014) dataset, for our list of ambiguous and unambiguous sentences. The MS COCO dataset comprises of a set of captions describing an image. Captions containing ambiguous verbs corresponding to 461 images (Ambiguous COCO; Elliott et al., 2016) constitute the ambiguous sentences for our experiment. 461 captions that were randomly sampled from MS COCO constituted the unambiguous sentences for the experiment.

**Complexity.** Corpus of Sentences rated with Human Complexity Judgments\(^1\) (Iavarone et al., 2021) and PACCSS-IT (Brunato et al., 2016) contain 20 human ratings on the scale from 1 (not complex) to 7 (very complex) about sentences. We binarize these ratings and consider sentences below the average to be simple sentences and others to be complex sentences. The resulting dataset is class-balanced (complex/simple) in terms of examples (592 sentences of each class for English and 551 sentences for Italian). The average sentence length for complex and simple examples is 24.84 and 13.95, respectively for English sentences. For Italian sentences, the average sentence length for complex and simple examples is 21.61 and 12.26, respectively. The complexity could therefore be encoded solely in the sentence length.

**Grammaticality.** For experiments under this category, we use the Benchmark of Linguistic Minimal Pairs (BLiMP; Warstadt et al., 2020) and the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2019) datasets. BLiMP contains sentence pairs, one of which contains a mistake in syntax, morphology, or semantics while the other is correct. The dataset covers 67 different conditions, grouped into 12 phenomena. These phenomena are further categorized as ‘syntax’, ‘morphology’, ‘syntax-semantics’ and ‘semantics’. The CoLA dataset is not contrastive but contains human annotations of acceptable grammaticality.

\(^1\)English sentences were taken from the Wall Street Journal section of the Penn Treebank. Italian sentences were taken from the newspaper section of the Italian Universal Dependency Treebank.
4 Experiments

4.1 Task definition

In the following experiments, we are solving three classification tasks in parallel. The input is always the whole sentence and the output one of the two classes (ambiguous/unambiguous, complex/simple, acceptable/unacceptable), as shown in Table 1, applies to the whole sentence. The whole pipeline is also depicted in Figure 1. When using the TF-IDF feature extractor, it replaces the pre-trained LM block.

![Figure 1: Example of the experiment pipeline for ambiguity classification. Ambiguous sentence from Stanley and Gendler Szabó (2000).](image)

4.2 Setup

We use a simple MLP classifier to identify three linguistic traits from BERT (bert-base or multilingual bert-base) and GPT-2. The resulting vectors are 768-dimensional.\(^2\) Both of these models are Transformer based models and contain 12 layers, which makes comparison convenient. We perform probing on each model separately.

- **CLS**: single vector at the [CLS] token.
- **Pooling**: single vector from the pooling layer.
- **Tokens**: vector representations of tokens aggregated with mean or (Hadamard) product to get a single 768-dimensional vector.

We obtain the layer-wise pre-trained model representations using Huggingface (Wolf et al., 2019) and use them to train a classifier that identifies if a sentence belongs to the positive class (e.g. ambiguous) or not. We perform a 10-fold cross-validation each with 10 runs of MLP.

**Baseline.** The most common class classifier (50% accuracy) is a poor baseline because it may be that the ambiguous and non-ambiguous sentences are distributed differently w.r.t. topic. In an attempt to alleviate this issue, we, therefore include as the baseline a TF-IDF-based vectorizer (with a varying number of maximum features). Probe performance of e.g. 65% would be considered at the first glance a positive result compared to 50%. However, in reality, it would be a false positive finding if a simple lexical feature extractor such as TF-IDF could yield 70%.

**MLP Configuration.** For probing we use MLPClassifier from scikit-learn 1.1.0 (Pedregosa et al., 2011) with most defaults preserved, as shown in Table 3.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Single hidden layer (100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation</td>
<td>ReLU</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate</td>
<td>(10^{-3})</td>
</tr>
<tr>
<td>Epochs</td>
<td>Early stopping, patience 1</td>
</tr>
</tbody>
</table>

Table 3: MLP classifier configuration.

4.3 Ambiguity & Complexity

Because the dataset is in Italian, we make use of multilingual BERT for both Complexity datasets. The probe performance for M-BERT is shown in Figure 2. At the first glance, it appears that the model does represent ambiguity internally since the ambiguity probe is systematically higher than 50%. However, because TF-IDF performs similarly and only uses surface-level features, the probe is very weak. This is supported by the fact that the most negative tokens from the classification (extracted from logistic regression coefficients) contained words such as *man* or *woman*, which disambiguate, based on gender, some unclear cases with an unclear referent.

In contrast, the complexity probe is systematically higher than the TF-IDF baseline. With minor exceptions, the accuracy remains high regardless of the layer. The performance for Italian (sentences taken from PACCSS-IT corpus) is identical to that for English using M-BERT (not shown). The CLS representation at layer 0 is 50% in both instances because it does not contain any information from the sentence (before the self-attention block).

4.4 Grammaticality

For the morphological task of determiner-noun agreement, Figure 3 shows a sudden drop in accuracy for the CLS representation at the 5th layer. In all the tasks concerning “Determiner-Noun Agreement”, the sentence minimal pairs focus on the number agreement between the demonstrative determiners (like this/these) and an associated noun.
Examples of minimal pairs from the different tasks of this kind are shown in Table 4.

While the cause is unclear, it corresponds to the average norm of the representation being very low at that particular layer, making it harder for the classifier optimization.

As Figure 4 shows, many tasks can be “solved” with a simplistic TF-IDF featurizer, making them inadequate for determining the usefulness of large model representations. More adequate datasets need to be developed for probing stronger models. Systematically for all cases in morphology where the TF-IDF failed to work accurately, the performance of CLS representations was worse than the mean representations. Even in most semantics tasks, TF-IDF probes had near-perfect accuracy. For the 7 out of 26 syntactic tasks where the TF-IDF classifier was not accurate, the BERT models show a steep rise in accuracy from the 2nd/3rd layer for the mean and CLS representations, respectively. In comparison, GPT-2 does not exhibit this pattern.

5 Discussion

The experiments with ambiguity reveal that the representations of the pre-trained models do not encode the ambiguity trait well. The description detailing how the Ambiguous COCO was created (Elliott et al., 2017) states that the dataset was created with the intention of testing the capabilities of multimodal translation systems. We posit that ambiguity as a trait is not encoded in an accessi-
For BLiMP tasks related to morphology and syntax-semantics, the accuracy goes down in the upper layers, presumably because of increasing abstraction for both models (not shown in graphs). Although we perform experiments without fine-tuning, the findings are in line with the experimental results of Mosbach et al. (2020) where finetuning on 3 tasks from the GLUE benchmark (Wang et al., 2018) showed changes in probing performance mostly in the higher layers. Fine-tuning however led to modest gains. The present setup which probes sentence representations from pre-trained models shows that the middle layers fare far better in our probing tasks than the upper layers. This leads us to posit that the features of interest are highly localized and are lost in the upper layers (even with fine-tuning).

Although both BERT and GPT-2 employ the Transformer (Vaswani et al., 2017) architecture, they have very different ways and locations for storing knowledge in their internal representations (Rogers et al., 2020; Vulić et al., 2020; Lin et al., 2019; Kuznetsov and Gurevych, 2020; de Vries and Nissim, 2021; Liu et al., 2021). The CLS representations outperform the mean representations in only a few cases. This is expected since without fine-tuning the CLS token in BERT is trained to be used for the next sentence classification tasks.

6 t-SNE Inadequacy

Given appropriate optimization and classifier, if two or more classes in a vector space form clusters, they are linearly separable and therefore the classifier performs well. Furthermore, if a classifier probe performs well and is not affected by surface-level phenomena, it means that the features are represented in the vectors. Both these statements are one-way implications:

- clear clustering $\rightarrow$ high classifier accuracy
- high classifier acc. $\rightarrow$ feature present
Because t-SNE projects vectors from high dimensional space to lower dimensions in a manner that tries to preserve distances, it may be that visual clusters are created where there were none before and vice versa. The following scenarios are possible:

- clear clusters and high classifier accuracy
- no clusters and high classifier accuracy
- no clusters and low classifier accuracy

The last combination, “clear clusters and low classifier accuracy” is impossible with proper optimization. The three scenarios on probes from the previous experiments are shown in Figure 5. The conclusion is that probes should always precede visual clustering checks using t-SNE because it may be that the data does not form clear clusters in t-SNE but the classes are still linearly separable, meaning that the feature is encoded. The last image shows two clusters but not those that separate the two classes.

A plethora of work uses t-SNE to show clusters of vectors grouped by features (Chi et al., 2020; Nigam et al., 2020; Wu et al., 2020; Zhang et al., 2021; Subakti et al., 2022), though some follow-up with reporting classifier performance. Because t-SNE visual separation is not easily quantifiable, the negative results are often underreported (Fanelli, 2012; Mlinarić et al., 2017). This issue can be resolved by using other methods, such as probes.

**Algorithm 1 Forcing t-SNE Clusters**

```
▷ Vectors of sentences in the two classes
Load $D_A$, $D_B$

▷ Cluster size, e.g. $|D_A|/2$
Input $c', c ← c'/2$

▷ Two seeds from classes, most distant
$s_A, s_B ← \arg \max_{v_A \in D_A, v_B \in D_B} ||v_A - v_B||$

▷ Closest points to own seeds
$C'_A ← \text{top-}c v \in D_A - ||s_A - v||$
$C'_B ← \text{top-}c v \in D_B - ||s_B - v||$

▷ Furthest points to opposing seeds
$C''_A ← \text{top-}c v \in D_A ||s_B - v||$
$C''_B ← \text{top-}c v \in D_B ||s_A - v||$

$C_A ← C'_A \cup C''_A$
$C_B ← C'_B \cup C''_B$
t-SNE($C_A \cup C_B$)
```

6.1 Forcing t-SNE Clusters.
It is possible to start with sentence vectors that result in a t-SNE graph that does not show any visual clusters and select half of them such that running t-SNE will show clusters between the two classes. The algorithm is described in Algorithm 1. It is based on first finding two most distant “seeds” from the two classes and then selecting vectors of the same class which are closest to the seed or most distant to the other seed.

An example is shown in Figure 6. While the original does not show any clusters between the classes, the application of the algorithm selects such vectors that t-SNE shows visual clusters. Simply randomly subsampling the vectors would not work but this shows that using t-SNE to visually determine the presence of a feature is not robust.

7 Conclusion
In this work, we showed how large pre-trained language models represent sentence ambiguity in a much less extractable way than sentence complexity and stress the importance of using reasonable baselines. We document that template-based datasets, such as BLiMP used for sentence acceptability, are not suitable for probing because of surface-level artefacts and more datasets should be developed for probing more performant models. Finally, we discuss why using t-SNE visually for determining whether some representations contain a specific feature is not always a suitable approach.

Future work
Because both t-SNE clustering and classification (inability to establish a rigid threshold for accuracy) can fail for determining whether a specific feature is represented in the model, more robust methods for this task should be devised. These probes should also be replicated in models used for machine translation, which is the primary motivation for studying these traits.

8 Acknowledgements
This work has been funded from the 19-26934X (NEUREM3) grant of the Czech Science Foundation. The work has also been supported by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2018101 LINDAT/CLARIAH-CZ.
References


Gerry Altman. 1985. The resolution of local syntactic ambiguity by the human sentence processing mechanism. In Second Conference of the European Chapter of the Association for Computational Linguistics.


Daniele Fanelli. 2012. Negative results are disappearing from most disciplines and countries. Scientometrics, 90:891–904.


Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel R Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *ACL (2)*.


Amber Nigam, Shikha Tyagi, Kuldeep Tyagi, and Arpan Saxena. 2020. Skillbert: “skilling” the bert to classify skills!


