A Deep Learning Perspective on Linguistic Ambiguity

Laura Aina

TESI DOCTORAL UPF / year 2021

THESIS SUPERVISOR
Gemma Boleda
Department of Translation and Language Sciences

upf. Universitat
Pompeu Fabra
Barcelona

Acknowledgements

I am immensely grateful for all the guidance and support I have received during the journey eventually leading to this thesis.

I want to express my deep gratitude to my supervisor Gemma Boleda, who has helped me in the last four years in so many ways. I am thankful, in the first place, for the precious opportunity of doing a PhD under her supervision, and for everything that came with it: the stimulating discussions, the practical advice, the "Ànims!", and the empathy, especially in the more difficult moments.

I am very grateful to the researchers that collaborated with me on the work presented in this thesis: besides my supervisor, Thomas Brochhagen, Kristina Gulordava, Xixian Liao, Tal Linzen, and Matthijs Westera. I have learned so much from working with all of them, and I want to thank them for believing in my ideas and helping me concretize them. I owe an especially big debt of gratitude to Thomas and Kristina, who devoted much time to helping me develop my research and navigate the general PhD process. This thesis could not have been here without their inspiring guidance and encouragement.

At Pompeu Fabra University, I have met brilliant researchers who contributed to the development of my research through frequent feedback and discussions, for which I am very grateful. In addition to those I have already mentioned: Marco Baroni, Roberto Dessì, Eleonora Gualdoni, Andreas Mädebach, Louise McNally, Carina Silberer, Ionut-Teodor Sorodoc, and Lucas Weber. I want to thank all the friends I have made in my department, and who have been amazing companions of this journey: the already mentioned PhD students from the COLT group, plus Erendira Cervantes, Alice Cravotta, Zi Huang, Alexandra Navarrete, Dominika Slušná, Raquel Veiga, Kata Wohlmuth, and Giorgia Zorzi. Thank you for all the help and the great time together. A special thanks to my "academic twin" Ionut, with whom I have shared the experience of the PhD from the very first day. Even the most difficult aspects became more bearable when we shared them and knew we were there for each other. I am deeply grateful to Ionut for all of his support and advice.

In early 2020, I had the opportunity of visiting the Department of Cognitive Science at Johns Hopkins University. I want to thank Tal Linzen for making this possible and for his support during our collaboration. Despite the pandemic complications, the research visit enriched my PhD journey with many opportunities for growth. I want to thank Suhas Arehalli, Najoung Kim, Tom McCoy, Karl Mulligan, Grusha Prasad, and Natalia Talmina for all the interesting discussions, and for making my time in Baltimore really special. I had to leave earlier than planned but it was enough time to come back home with more friends to count on.

In the course of the PhD, I was lucky to have the chance of discussing research with many people, including planning potential collaborations that unfortunately did not get to happen. In particular, I want to thank Marianna Bolognesi, Marco Del Tredici, Raquel

Fernández, Aina Garí Soler, Sandro Pezzelle, Matthias Unterhuber, and Sina Zarrieß. I also want to thank Kate Aspinall and Kat Mustatea for the inspiring discussions about technology and art.

I want to express my gratitude to the members of the board evaluating this thesis: Allyson Ettinger, Alessandro Lenci, and Hannah Rohde. They have inspired me with their research and I feel honored that they agreed to read and evaluate mine.

The research I present in this thesis was made possible thanks to 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 Catalan government (SGR 2017 1575), and the computer resources at CTE-POWER from the Barcelona Supercomputing Center (RES-IM2019- 3-0006; RES-FI-2018-3-0034).

On a personal note, I have many people I want to thank for their invaluable help. They endured my rambling about the PhD, encouraged me, made me laugh, and gave me a hand when I needed it. Without all this, I could not have written this thesis, and I feel immensely lucky to have these people in my life. Besides the friends I have already mentioned: Erica, Dipa, Lisa, Livia, Magda, Mari, Maria Chiara, Natascia, Rombo, Rosa, Veronica, Viola, and Zoi. A special thanks to Livia, Natalia, and my sister Chiara, who patiently listened and helped me figure things out on a daily basis throughout this whole process. Finally, to my family and Luis: thank you for the countless ways you have been supporting me, and for believing in me, especially when I don't do it enough. In this journey, with all the twists and turns, my highs and lows, you have been a constant presence keeping me strong. Grazie, gracias.

Abstract

This thesis investigates what mechanisms humans and artificial systems use to deal with linguistic ambiguity. I adopt computational linguistics, in particular deep learning methods, as my research framework, and focus on the English language. Two studies investigate how neural language models – trained on unlabeled text corpora – process lexical and syntactic ambiguities, by inspecting both the internal representations and predictions of these models. The findings indicate both effects of default preferences over interpretations and a good level of context-sensitivity, though with room for improvement. In two other studies, I test linguistic hypotheses on large-scale corpus data, by deriving computational estimates of contextual expectations about word-level content and reference, respectively, through deep learning models. The results suggest that 1) lexical disambiguation relies on a context-dependent interplay between the information contributed by an expression and its context, and 2) speakers tend to balance the informativeness of a referring expression and that of its context.

Resum

Aquesta tesi investiga quins mecanismes utilitzen els humans i els sistemes artificials per respondre a l'ambigüitat lingüística. Adopto la lingüística computacional, i en particular els mètodes d'aprenentatge profund ("deep learning"), com a marc de recerca, i analitzo dades de l'anglès. En dos dels estudis, investigo com els models de llenguatge neuronals, entrenats amb dades de corpus textuals sense etiquetar, processen les ambigüitats lèxiques i sintàctiques, inspeccionant tant les representacions internes com les prediccions d'aquests models. Els resultats indiquen tant efectes lèxics / sintàctics per defecte sobre les interpretacions com un bon nivell de sensibilitat al context, tot i que amb marge de millora. En els altres dos estudis, avaluo hipòtesis lingüístiques amb dades de corpus a gran escala mitjançant models d'aprenentatge profund. Els models proporcionen estimacions de les expectatives contextuals sobre la interpretació d'una paraula i sobre el referent d'una expressió, respectivament. Els resultats suggereixen que 1) la desambiguació lèxica es basa en la interacció entre la informació aportada per una expressió i el seu context, que varia segons els casos, i 2) els parlants tendeixen a trobar un equilibri entre la informativitat d'una expressió referencial i la del context en què s'utilitza.

Resumen

Esta tesis investiga qué mecanismos utilizan los humanos y los sistemas artificiales para responder a la ambigüedad lingüística. Adopto la lingüística computacional, en particular los métodos de aprendizaje profundo, como mi marco de investigación, y me centro

en la lengua inglesa. Dos de los estudios investigan cómo los modelos del lenguaje, entrenados a partir de corpus de texto sin etiquetar, procesan las ambigüedades léxicas y sintácticas, inspeccionando tanto las representaciones internas como las predicciones de estos modelos. Los resultados indican tanto efectos léxicos / sintácticos por defecto sobre las interpretaciones como un buen nivel de sensibilidad al contexto, aunque con margen de mejora. En los otros dos estudios, investigo hipótesis lingüísticas con datos de corpus a gran escala mediante modelos de aprendizaje profundo. Los modelos proporcionan estimaciones computacionales de expectativas contextuales sobre la interpretación de una palabra y sobre el referente de una expresión, respectivamente. Los resultados sugieren que 1) la desambiguación léxica se basa en una interacción entre la información aportada por una expresión y su contexto, que varía según los casos, y 2) los hablantes tienden a buscar un equilibrio entre la informatividad de una expresión referencial y la de su contexto.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

> *Your gaze scans the streets as if they were written pages: the city says everything you must think, makes you repeat her discourse, and while you believe you are visiting Tamara you are only recording the names with which she defines herself and all her parts. However the city may really be, beneath this thick coating of signs, whatever it may contain or conceal, you leave Tamara without having discovered it. Outside, the land stretches, empty, to the horizon; the sky opens, with speeding clouds. In the shape that chance and wind give the clouds, you are already intent on recognizing figures: a sailing ship, a hand, an elephant...*
>
> – Italo Calvino, *The Invisible Cities*, 1972

Ambiguity is entrenched in linguistic communication and implies that during comprehension expressions need to be interpreted; that is, we need to infer what our interlocutor intended to convey. This thesis investigates the mechanisms underlying linguistic ambiguity, through a computational linguistics perspective. In particular, I adopt the paradigm of deep learning as the main research framework, with the goal of 1) investigating the behavior emergent in an artificial system trained merely from exposure to language usage (a language model), and 2) testing linguistic hypotheses at scale by estimating information through data-driven methods.

1.1 Motivation

Natural languages allow for ambiguities to flourish. To give an example at the word level, the word “chair” can be used to refer to both the piece of furniture 🪑 (1a), or the head of an organized group 👩‍🏫 (1b).[1]

(1) a) I have bought a desk but I still need to get a *chair*.

b) She gladly accepted her new role as department *chair*.

[1] Throughout the thesis, I will often use emojis as a shortcut to refer to word senses.

The case of "chair" exemplifies **lexical** (semantic) **ambiguity** (Cruse, 1986; Rodd, 2018), and is far from being an isolated instance.[2] Dictionaries provide tangible evidence of this as they list more than one definition for most words.[3]

Ambiguity is not restricted to words and can affect various levels of linguistic analysis, from phonology to pragmatics. In this thesis, I focus on a subset of ambiguity types: the aforementioned lexical ambiguity, and the ambiguities exemplified by the sentence pairs presented below (the region subject to the ambiguity is in italics). The sentences in (2) are an example of **syntactic ambiguity** (Frazier, 1978; Pickering and Van Gompel, 2006). The parse of the sentence is initially ambiguous as at first "witness" can both serve as direct object of the main verb (2a) or introduce an embedded clause (2b).

(2) a) *The detective believed the witness* and continued her investigations.
b) *The detective believed the witness* was lying.

By contrast, (3) displays a case of **referential ambiguity** (Ariel, 1990; Nieuwland and Van Berkum, 2008), where the pronoun "she" could be referring to either of the introduced entities, but depending on the context it is more likely to refer to one or the other (Ann in (3a), and Betty in (3b)).

(3) a) Ann apologized to Betty, because *she* had broken her vase.
b) Ann congratulated Betty, because *she* published her novel.

What these sentence pairs overall exemplify is that the structure of natural languages – in this case, English – allows expressions to be subject to multiple interpretations. Speakers tend not to avoid this type of expressions (Ferreira, 2008; Wasow, 2015), for instance, because they are easier to produce (the most ambiguous expressions tend to be short and frequent; Zipf 1949). But we know from experience that linguistic communication is typically successful (i.e., interlocutors understand each other): we rarely remain indecisive about the meaning of an utterance or get misled to the wrong interpretation.

How is this possible, given the ubiquitous presence of ambiguity in language? Which mechanisms allow humans to successfully transmit information to each other when the code used to do so does not univocally determine the interpretation of expressions? This thesis addresses these general questions. I consider the aforementioned ambiguity types (lexical, syntactic, and referential) in English, and study strategies and factors enabling to infer the intended interpretations of expressions.

I focus on the interaction between two sources of information during the interpretation of an expression. First, we have the **expression** itself, in the sense of the set of potential interpretations that the expression is associated with. As users of a language,

[2] Lexical ambiguity may also lead to syntactic ambiguities when a word is associated with different morpho-syntactic categories (e.g., "hope" as noun or verb).

[3] According to Rodd et al. (2002), for at least 80% of common words a dictionary lists multiple senses.

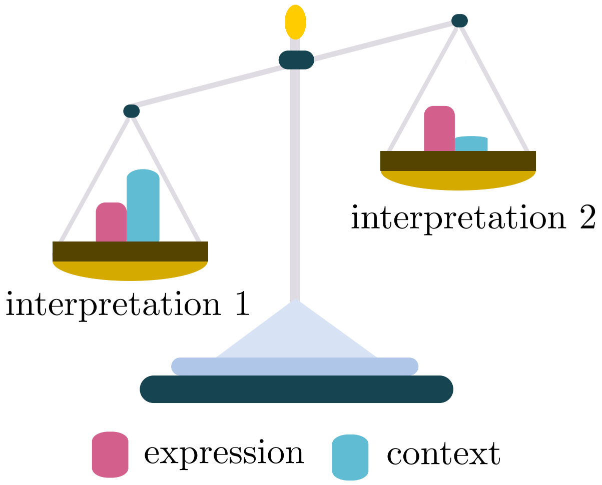


Figure 1.1: Metaphorical visualization of the interaction between an expression and its context during ambiguity resolution: both contribute a weight towards each potential interpretation.

we abide by conventions (e.g., from the grammar or the lexicon) about what could be meant by an expression. Because of ambiguity, multiple interpretations may however be viable. Some may be considered more prototypical, for instance, because they are more frequent (Duffy et al., 2001) or simple (Frazier and Fodor, 1978). This may cause that some interpretations are by default preferred. For example, upon hearing "chair" without any context, the 🪑 sense probably first comes to mind.

But expressions always occur in a specific situation. As we saw in the sentence pairs (1-3), a **context** evokes preferences over what content is more relevant as an interpretation of an expression. This may be due to both language-internal factors (e.g., syntactic constraints, selectional preferences) and the plausibility of the state of affairs that the speaker might be intending (e.g., world knowledge about events, consistency with information in the common ground). These contextual expectations will interact with the information from the expression, including potentially overriding the a priori (out-of-context) preferences over its interpretation.

In a simple scenario with two potential interpretations, we can thus envision the final state of an ambiguity resolution process as the equilibrium reached in a double-pan balance (Figure 1.1). Each pan is a potential interpretation; the expression and the context load each pan with a different weight, depending on the interpretation that they privilege. Metaphorically speaking, this thesis studies the mechanisms of such a weighing procedure. I investigate the information that both the expression and the context contribute to the interpretation of expressions, and their interaction in the successful resolution of ambiguity.

A part of my experiments focuses on the interpretation strategies that artificial systems learn from mere exposure to word usage at scale (a neural language model), while receiving no explicit training nor inductive bias as to how to resolve ambiguities: How successful are these models in accounting for an expression's contextual interpretation? Are they a priori biased towards some readings, and if so, can they correctly use the context to overcome those preferences when not panning out in context? In another group of analyses, I test linguistic hypotheses about the interplay between the information coming from the expression and from its context for interpretation: To which degree should the expression and its context, respectively, be relied on to recover the intended meaning? Do speakers strive to give as much information as possible to the addressee to ease their interpretation process, or only within the limit of efficient communication?

1.2 Approach

To carry out my research, I resort to the tools of **deep learning** (LeCun et al., 2015), a powerful machine learning paradigm to build computational models. A deep learning model considers an objective task; within Natural Language Processing (NLP), for instance, to predict the next word in a linguistic sequence – language modeling–, or if a set of expressions refer to the same entity – coreference resolution (Jurafsky and Martin, 2009). The output behavior is determined by computing a series of intermediate distributed representations and learned by spontaneously identifying patterns in the training data.

In the last decade, deep learning led to dramatic advances in NLP. This raised questions around ways in which this framework can constitute a resource for linguistic research (Manning, 2015; Linzen, 2019; Baroni, 2021). We can roughly cluster two main approaches. On the one hand, deep learning models can be used to automate the estimation of some linguistic information. For instance, a common application is to study the effects of predictability on human sentence processing, using word probability scores from language models (e.g., Goodkind and Bicknell 2018; Van Schijndel and Linzen 2018). But deep learning models can also themselves be analyzed in how they process and respond to linguistic phenomena (see Belinkov and Glass (2019) for an overview). This is not only a thorough way of evaluating these models from an NLP perspective. It also allows to investigate the conditions that enable the modeling of a linguistic phenomenon, helping to characterize it or individuate its responsible mechanisms in humans.

Building on both of these approaches, the way I use deep learning to investigate linguistic ambiguities is thus two-fold: 1) to study the inner dynamics of deep learning models, and 2) to expedite linguistic analyses by using deep learning models to estimate information. For each set of experiments reported (chapters 3 to 6) I introduce and apply novel methodologies, connected to either purpose.

Concretely, chapters 3 and 5 are concerned with the analysis of **neural language models** (Mikolov et al., 2010; Bengio et al., 2003; Devlin et al., 2019; Radford et al., 2018). These deep learning models output a probability distribution over words given a linguistic context: for instance, in "For dinner, I ate ..." nouns referring to food should be the most likely. These models are trained only by exposure to language usage in a text corpus. Yet, we can expect them to develop mechanisms to handle ambiguities in order to display the correct output behavior. My experiments aim to clarify the nature and success of these mechanisms. For this goal, I analyze both the internal representations and the output predictions of language models.

By contrast, in chapters 4 and 6, I use deep learning models as a tool to ease the testing of linguistic hypotheses on large datasets. Both comprehension and production strategies related to ambiguity are difficult to test at scale, considering a wide range of expressions and their occurrences in discourse contexts. Computational data-driven methods to estimate information can provide much more coverage and flexibility, reducing the need to collect human judgments. For this purpose, in Chapter 4, I use again language models, in particular, to represent word-level expectations and the lexicon. In Chapter 6, I instead model reference-level expectations through a system trained to predict an entity given a context, through a variant of the coreference resolution task.

1.3 Research Goals

This thesis contributes to research in linguistics, cognitive science and NLP, by presenting a comprehensive study of linguistic ambiguity considering different angles of investigation and ambiguity types.

Concretely, I address the following research goals:

- **Clarifying how ambiguities are processed by neural language models**
  In particular, I aim to establish:

  - The degree of responsiveness of language models to contextual cues revealing the intended interpretation of an expression;
  - The effect of default (out-of-context) preferences over the interpretation of an expression.

  I investigate these aspects on both lexical and syntactic ambiguities (chapters 3 and 5). These studies act as an evaluation of these models (assessing which aspects of ambiguity can be successfully learned from data and those that instead pose challenges) and a clarification of their inner workings. The behavior of language models, as observable through the experiments, is compared with what is know about how humans interpret expressions.

- **Studying the interaction between the information contributed by an expression and its context, respectively, in ambiguity resolution**
  I investigate this aspect from both the comprehension and production perspectives, by assessing the following:

  - The degree of reliance on contextual expectations or an expression that best models the resolution of ambiguities;
  - To which extent the potential challenges of ambiguity are taken into account by speakers during production.

  In Chapter 4, I develop a computational framework to model and test lexical ambiguity resolution as the combination of information from the lexicon and contextual expectations. In Chapter 6, I analyze the relation between the expression chosen to refer to an entity and the informativeness of its context (i.e., how predictable the referent is).

1.4 Structure of the Thesis

Chapter 2 I provide an overview of research on ambiguity in linguistics and cognitive Science. I then introduce deep learning models in the context of NLP and summarize what we know about their processing of ambiguities.

Chapter 3 I study the resolution of lexical ambiguity in neural language models. Using supervised auxiliary tasks, I extract and evaluate the word information captured in the internal representations of the models.

The research presented in this chapter is partially based on the following paper:

- Laura Aina, Kristina Gulordava, Gemma Boleda. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the ACL 2019 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Chapter 4 Modeling different interactions between the lexicon and contextual expectations (represented using language models), I study the degree to which the two sources of information are relied on for lexical ambiguity resolution.

The research presented in this chapter is based on the following papers:

- Kristina Gulordava, Laura Aina, Gemma Boleda. How to represent a word and predict it, too: improving tied architectures for language modelling. In *Proceedings of the EMNLP 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

- Laura Aina, Thomas Brochhagen, Gemma Boleda. Modeling word interpretation with deep language models: The interaction between expectations and lexical information. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society (CogSci 2020), 2020*

Chapter 5 I study the way neural language models process temporary syntactic ambiguities. I use text generation to assess their default biases in lack of disambiguating cues, and how the behavior changes when these are given.

The research presented in this chapter is based on the following paper:

- Laura Aina, Tal Linzen. The Language Model Understood the Prompt was Ambiguous: Probing Syntactic Uncertainty Through Generation. In *Proceedings of the 2021 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2021.

Chapter 6 I investigate the relation between referent predictability and a referring expression's form (its syntactic type and length), using computational estimates of the former. There estimates are derived through a variant of the coreference resolution task.

The research presented in this chapter is based on the following paper:

- Laura Aina, Xixian Liao, Gemma Boleda, Matthijs Westera. Does referent predictability affect the choice of referential form? A computational approach using masked coreference resolution. In *Proceedings of the 2021 CoNLL Conference on Computational Natural Language Learning*, 2021.

Chapter 7 I discuss the contributions of the thesis, as well as directions for future research, jointly considering the insights from the various experiments reported.

Chapter 2

BACKGROUND

In this chapter, I discuss previous research relevant to the experiments reported in the thesis. In the first part of the chapter (Section 2.1), I provide an overview of the way linguistic ambiguity is accounted for within linguistics and cognitive Science, focusing on the ambiguity types I consider in my studies (lexical, syntactic, and referential). I then introduce deep learning models and summarize what we know about the way they resolve linguistic ambiguities, based on their workings and previous research (Section 2.2).

2.1 Linguistic Ambiguity

To clarify how ambiguity affects language, let us start from the basics of how linguistic communication proceeds. A **speaker** aims to transmit some information to an **addressee**. The speaker encodes the message with some linguistic form and the addressee has to decode the speaker's message. In the absence of ambiguities in language, it would be straightforward to determine what the interlocutor intended to convey, by relying on a one-to-one mapping between forms and meanings (i.e., each expression can only be understood in one way). But since ambiguities are often present, the comprehension of the utterance needs to include an **interpretation** step: to identify, among various options, the intended reading.

Ambiguities thus introduce uncertainty in linguistic communication. From the perspective of successful transmission of information, it thus seems compelling to consider them problematic. Indeed, formal languages such as programming ones are purposely designed to avoid ambiguity, with a univocal mapping between strings and operations to execute. Why natural languages developed to allow ambiguities is therefore puzzling and their pervasiveness calls for an explanation (Wasow et al., 2005). One perspective is that ambiguity is evidence that natural languages are not optimized for communicative efficiency (Chomsky, 2002). On the contrary, it has been argued that ambiguity offers functional advantages for a communicative system. For instance, Zipf (1949) claimed that ambiguity permits a balance in minimizing the efforts of both the speaker and the addressee: in the extreme case, the speaker's effort is minimized if there is only one word in the language, while the addressee's effort is minimized if each word is assigned a distinct meaning. Piantadosi et al. (2012) further developed this idea and claimed that ambiguity achieves a trade-off between clarity and ease: it makes a language compact, re-using forms that are easier to produce or comprehend (short, frequent, and phonotactically well-formed). This does not obstruct communication as long as ambiguous expressions get used in informative contexts, which allow recovering the intended meaning.

But what are actually the mechanisms that make us capable of swiftly and successfully resolving ambiguities? These mechanisms are typically described through separate linguistic accounts for ambiguities at different levels (e.g., lexical, syntactic, etc.). Before I discuss these accounts, I below provide an overview of the main dimensions used to characterize ambiguities and their resolution.

What are the potential interpretations of an expression? This depends on the level of analysis that the ambiguity involves. With respect to syntax, the interpretations of an expression are strictly constrained by the grammar of the language (Pickering and Van Gompel, 2006); they can be codified as (sub)trees, syntactic roles – e.g., direct object – or morpho-syntactic categories – e.g., noun. Within lexical semantics/pragmatics, one typically focuses on *senses*, which may be consolidated in the lexicon – e.g., "palm" as 🌴 and ✋ –, or ad hoc to a situation – e.g., "Her words are *bullets*" (Falkum and Vicente, 2015). At the referential level, the interpretations are the set of real-world entities an expression could refer to, which can be contextually constrained (e.g., the entities in the discourse or visual context; Nieuwland and Van Berkum 2008).

What is (out of context) the contribution of an expression? Abstracted from any context, an expression has a set of potential interpretations. It may be equally associated with each of these, or alternatively, exhibit a bias towards a subset. Accounts of ambiguity may posit that expressions come with default preferences over interpretations, due to, e.g., frequency. These preferences can be taken, for instance, to enforce an interpretation in neutral contexts, to make some readings harder to derive, or to determine the information initially activated during processing (Frazier, 1978; Duffy et al., 1988).

How does a context affect interpretation? The context can create a bias towards a reading of an expression through different types of cues. For instance, in (4), "suit" is disambiguated to its noun or verb reading by the following context. The responsible sentential cues are rigid (i.e., no other interpretation is possible) and purely linguistic (driven by grammar; Frazier and Rayner 1987).

(4) The pants *suit* are great. vs. you well.

By contrast, in (5), the preceding discourse context (beyond the sentence boundaries) induces a preference (potentially defeasible) for the sense of "toast", which seems the most congruent with what has been said so far (considering the interaction of the linguistic context with world knowledge; Wilson and Carston 2007).[1]

(5) a) Finally, Beth got the promotion. She proposed a *toast*. 🥂

b) Bob asked Beth what he could cook. She proposed a *toast*. 🍞

What is the time-course of ambiguity resolution? As disambiguating cues can occur before or after an ambiguous expression, one can contrast the behavior under different types of preceding contexts (e.g., Frazier and Rayner 1987; Foraker and Murphy 2012). If the context is neutral (no disambiguating cues; e.g., "The *palm*..."), the addressee may leave the interpretation open until disambiguating cues are given (e.g., "The *palm* of my hand"), or commit to an interpretation, subject to revision if ultimately incorrect. In disambiguating prior contexts (e.g., "She grew a *palm*..."), the addressee may directly engage with the supported interpretation, or initially activate all to then select one.

Do speakers optimize their utterances to ease interpretation? To facilitate linguistic communication, speakers may optimize their utterances to reduce the interlocutor's challenges with ambiguity (Grice, 1975; Piantadosi et al., 2012). They may purposely avoid the use of ambiguous expressions, when not clear by the context how they would have to be resolved (Ferreira et al., 2005). But there may be also an opposite incentive to use ambiguous expressions: they tend to be more frequent and short than more transparent expressions (e.g, "it/mouse/computer mouse"), allowing the speaker to reduce their production cost.

2.1.1 Lexical Ambiguity

Lexical ambiguity broadly corresponds to the circumstance whereby a word form has a multiplicity of potential interpretations, or *senses* (Cruse, 1986; Small et al., 2013; Rodd, 2018). It is however a multi-faceted phenomenon, varying, in particular, in the relations among the potential senses of a word.

An important dimension in this respect is the degree of **relatedness** of word senses; that is, their conceptual overlap. *Polysemy* involves ambiguity among related senses: for instance, "glass" can refer to the material, a container of that material, or its content (6). Several sense alternations are regular across words in the lexicon, while others are word-specific (Apresjan, 1974).

[1] The contextual preference is defeasible by the following discourse context, without leading to a contradiction: for instance, in (5b) Beth may propose a toast 🥂 to celebrate that Bob finally offered to cook for her.

(6) a) The goblet is fragile: it is made of *glass*. 🍷 Material
b) She was relaxing on the sofa when the *glass* fell. 🍷 Container
c) I didn't expect her to drink the whole *glass*. 🍷 Content

By contrast, other words, like "bow" (7), are associated with unrelated interpretations, potentially even due to disjoint etymologies. These cases – the minority within lexical ambiguity – are described as *homonymy*.[2]

(7) a) The gift came wrapped and with a *bow*. 🎀
b) The archer firmly held the *bow* before shooting the arrow. 🏹
c) The passengers moved to the *bow* of the boat. 🚢

The interpretations of a word may also differ in **morpho-syntactic category**. These cases of ambiguity interact with relatedness, and may involve interpretations that share a conceptual basis (as in "smile"; (8)) or not at all (as in "bear"; (8)).

(8) a) You have a beautiful *smile*. 🙂 Noun
b) The surprise is going to make you *smile*. 🙂 Verb

(9) a) The polar *bear* is a vulnerable species. 🐻 Noun
b) The shelf could not *bear* the weight of the books. Verb

If multiple categories are possible for a word, lexical ambiguity thus also gives rise to a syntactic ambiguity (see next section). Cross-categorial polysemy is particularly frequent in English, as, for instance, denominal verbs can be formed by simply using the noun root as verb (Clark and Clark, 1979).

Finally, senses of a word may vary in their degree of **conventionalization** (Carston, 2021). Some may be more common and thus consolidated in the lexicon of a language (to the point they are, for instance, included in a dictionary). But word meaning is flexible and dynamic (Ludlow, 2014): the interpretation of a word can deviate from the typical senses, leading to convey ad hoc nuances in each context (11-10).

(10) Her brain is a *sponge* for knowledge. ≠ 🧽

(11) The plastic bag *dances* in the wind. ≠ 💃

Considering all these facets, it is clear that lexical ambiguity can affect, in one way or another, essentially all words in a language, though typically it is resolved effortlessly in communication. The mechanisms that enable this pose several research puzzles.

Accounts within cognitive science and psycholinguistics focused on the information that is activated during the processing of a word – *word meaning access* (see Frisson

[2] In this thesis, due to focusing on text data, I consider as ambiguous all homographs, including heteronyms – forms that have different meanings depending on the pronunciation (e.g., the case of "bow"). While these are not ambiguous in speech, they are ambiguous in text.

(2009) and Rodd (2018) for overviews). Experimental evidence tends to support an *exhaustive access* view, where information relevant to multiple interpretations of a word – whether disjoint or based on a common core – is activated when the word is processed (Duffy et al., 1988; Frisson and Pickering, 1999; Rodd et al., 2010). This occurs rapidly, with the person typically remaining unaware of the process. This information about different senses, though accessed in parallel, may not be equally activated. Both the dominance of a word sense (typically, codified as its frequency) and the extent to which the preceding context supports it tend to affect its activation (Rayner and Frazier, 1989; Duffy et al., 1988; Rodd, 2020). After this initial access to multiple senses, one tends to be rapidly selected (Tanenhaus et al., 1979; Rodd et al., 2005), suppressing the activation of non-selected senses. If the preceding context does not provide sufficient evidence to select an interpretation, two strategies are in principle possible: The commitment to a sense is delayed until disambiguating cues in the following context are integrated. Alternatively, a sense is selected – for instance, based on sense dominance – to be later revised at need. The chosen strategy seems to depend on the relatedness of the word senses (Frazier and Rayner, 1990): Senses with little conceptual overlap prompt a rapid commitment, likely due to being mutually incompatible. On the contrary, the choice among senses that share a common core can be delayed.

Linguists have proposed various explanatory accounts of the mechanisms allowing to interpret words in context. A first subject of debate is how information about a word is organized in the mental lexicon (our memory storage of word knowledge). Following Falkum and Vicente (2015), we can group approaches as instantiating either a *sense enumeration* or an *underspecification* view. In the first case, the different senses of a word are separately represented and stored (Klein and Murphy, 2001). In the second case, a unique underspecified representation is associated with a word, encompassing information relevant to various senses: e.g., a rich structured representation (e.g., Pustejovsky 1991), or a pointer constraining what the word may convey (Carston, 2012). An underspecification account can better model the productive aspects of polysemy, as senses get generated from the lexicon, rather than selected among a predefined set. It has however been argued that the sense enumeration view better suits homonymy (Klepousniotou, 2002). Some proposals accommodate differences due to sense relatedness, but treat this as a continuous spectrum, instead of focusing on the polysemy-homonymy dichotomy (Rodd et al., 2002).

As for the posited disambiguation mechanisms, approaches tend to describe interpretation as either dependant on lexicon-internal processes, or as the result of on-line inferences (Falkum, 2015). The former group of approaches focuses on interactions between the words in a sentence; for instance, leading to apply lexical rules (Asher and Lascarides, 2003), or with the words mutually modulating their meaning (i.e., *co-composition*; Pustejovsky 1995). These mechanisms tend to account well for the most regular aspects of polysemy. The latter group of approaches takes disambiguation to be a pragmatic matter and broadens the influence of context to the discourse and extra-linguistic information (Recanati, 2004; Wilson and Carston, 2007). For instance, (5b) is a case where the global context overrides preferences from the local sentential context (Cosentino et al., 2017): “to propose a toast” typically refers to an invitation to drink, but the discourse suggests another reading. A prominent lexical pragmatic approach is that of relevance theory (Wilson and Sperber, 2006): Disambiguation is achieved by adjusting the concept associated with a word, through mechanisms like narrowing or broadening. The process settles when meeting contextual expectations of relevance (i.e., a concept that seems plausible to be intended). This general reasoning strategy is taken to account for a broad range of phenomena up to one-off idiosyncratic usages of words (e.g., the examples in (11)-(10)).

2.1.2 Syntactic Ambiguity

In a syntactic ambiguity, multiple parses of a sentence or portion of it are simultaneously possible (Pickering and Van Gompel, 2006; Clifton Jr. and Staub, 2008). In a *globally* ambiguous sentence, the entirety of the sentence is subject to syntactic ambiguity, with uncertainty about its structure and consequently meaning.

(12) The friend of your colleague with blonde hair is coming to the party.

In (12), for instance, it is unclear where the prepositional phrase “with blonde hair” attaches, and thus whether the friend or the colleague has blonde hair. By contrast, in a *locally* ambiguous sentence, only a part of the sentence could have different interpretations: in the context of the sentence, the intended parse is clear.

(13) a) *Jane forgot the cake* in the oven.

b) *Jane forgot the cake* had to be frosted.

The main verb “forgot” takes “the cake” as noun phrase complement (direct object) in (13a), and a sentential complement, whose subject is “the cake”, in (13b). Other examples are lexical syntactic ambiguities as in (8-9). Compared to, for instance, polysemy, the syntactic interpretations of an expression are not flexible, as they are strictly constrained by grammar.[3] For analogous reasons, the correct disambiguation of a locally ambiguous sentence directly follows from the application of grammatical knowledge. For instance, in (13) the verb “had” rules out the direct object interpretation of “cake” which instead has to be the subject of “had” (else the parse would be ungrammatical).

Due to the topic of my experiments (Chapter 5), I here in particular focus on the case of **temporary syntactic ambiguities**, the type exemplified by (13): in these cases,

[3] An exception are innovative cross-categorial usages of words; e.g., “to chopstick” (Clark and Clark, 1979; Carston, 2019)

the local ambiguity affects the initial sentence portion, but, as the sentence unfolds, the parse becomes unequivocal (Frazier, 1978). Sentences as (13b) tend to elicit a so-called *garden-path* effect (Bever, 1970): a high processing cost at the disambiguating region (from "had" onwards) in comparison to the unambiguous version of the sentence ("Jane forgot *that* the cake..."). This effect has been construed as caused by an initial preference for the ultimately incorrect parse (the transitive reading of "forgot" in (13b)). Several accounts have been proposed in psycholinguistics to account for this behavior. Beyond modeling garden-path effects, temporary syntactic ambiguities proved useful to investigate how ambiguities are in general resolved in the course of incremental processing.

First, one can investigate which interpretation is by default – initially – favored, to which degree, and on the basis of which factors. Different parsing strategies have been proposed. For instance, the *garden-path model* by Frazier and Fodor (1978) proposes that when a sentence is compatible with multiple parses, one is immediately selected based on purely syntactic principles (such as minimal attachment): we only consider one parse at a time and in particular the simplest available. Constraint-based accounts instead posit that syntactic ambiguities are resolved somewhat analogously to lexical ambiguities (MacDonald et al., 1994; Trueswell and Tanenhaus, 1994; McRae et al., 1998): Multiple interpretations can be, at least to some degree, available, taking into account factors that affect their plausibility. This may come from the preceding discourse context, or from the ambiguous expression itself. Regarding the latter, in sentences like (13), aspects like the bias of "forgot" for a transitive reading and the thematic fit of "the cake" as its argument were shown to play a role (Garnsey et al., 1997), with an interaction between syntactic and lexico-semantic factors in the initial ambiguity resolution.

Within both accounts, garden-path effects arise due to having to reconsider the parse of the sentence (Pritchett, 1988): the initially preferred parse – whether the only one conceived or just the most plausible – is contradictory with evidence in the sentence. Some accounts studied the nature of this reanalysis process: whether it involves reprocessing the sentence, or a repair strategy attempting to preserve as much as possible the current representation of the sentence (Sturt et al., 1999; Grodner et al., 2003). Some studies also found that the reanalysis process is not always successful. Effects of the initial misinterpretations may linger even after disambiguation (Christianson et al., 2001). In a sentence like "While Anna dressed the baby spit up on the bed", subjects were found to incorrectly assign thematic roles, holding that the baby got dressed, and not Anna. Evidence of this kind led to challenging the idea that the language system always derives accurate and complete representations of utterances (Ferreira et al., 2002; Ferreira and Patson, 2007). Rather, these may be superficial in absence of pressure for a detailed understanding – hence just "good enough": Ambiguities may not be resolved, or the interpretation taken may not be consistent with the available evidence.

Under an alternative account, garden-path effects are not due to reanalysis but rather predictability (Hale, 2001; Levy, 2008, 2013). Surprisal theory takes comprehension to be expectation-based, with predictable events being facilitatory for processing. Applied to syntactic processing, it posits a fully parallel parser, where each potential sentence structure (and thus interpretation) is assigned some probability. Processing difficulties at disambiguation then arise out of a general low predictability effect: the parse that turns out to be correct was initially not deemed likely. This hypothesis was empirically investigated with the tools of probabilistic language models, computing probabilities of words or syntactic structures. These include syntax-aware models such as probabilistic grammars (Hale, 2001), as well as neural language models trained solely from text corpora (Van Schijndel and Linzen (2018); see Section 2.2 for an introduction to these models). Predictability was found to account well for the existence of garden-path effects, but less so for the magnitude of the disambiguation difficulties across constructions (Van Schijndel and Linzen, 2021).

2.1.3 Referential Ambiguity

An essential property of language is that it allows us to refer to things in the world: Linguistic expressions are not only associated with some conceptual information they evoke but they also link to objects in our reality (Frege, 1892). To infer what an expression refers to is fundamental to language comprehension, but ambiguities may occur also at this level if an expression could potentially refer to more than one entity.

(14) Bill told James the secret because it was important to *him*.

For instance, in (14), is the secret important to Bill or to James? Both options could in principle hold but a combination of semantico-pragmatic and grammatical factors can guide us to pick a referent (Hobbs, 1978; Grosz et al., 1995; Kehler and Rohde, 2013).

Due to the focus of this thesis (in particular of Chapter 6), I here focus on the interaction between ambiguity and **reference production** – the form chosen to refer to an entity. When a speaker plans a referring expression, they are faced with several options (Davies and Arnold, 2019). Suppose someone wants to refer to me: they can call me by my name (“Laura”, “Laura Aina”), put together a description (“the student”, “the PhD student from Pompeu Fabra University”), or use a pronoun (“you” or “she” depending on whether I am the interlocutor). A proper name acts as a rigid label directly linking to an entity. A description, by means of a full noun phrase, aids the identification of the referent by illustrating one of its properties. A pronoun instead provides little informational content but rather acts as a referential device: it points to an entity in the context. These options vary in how much they convey about the referent; that is, their *informativeness*, or *explicitness* (Ariel, 1990). Consequently, an expression may be more or less ambiguous: If two female entities have been introduced, “she” could refer to either. The same goes for “the student” if multiple students are part of the context.

Various accounts of the mechanisms driving the choice of a referring expression have been proposed. Differences aside, these proposals generally point to an interaction between an expression and its context of use (Arnold and Zerkle, 2019). A prominent family of accounts takes each entity introduced in a discourse context to be associated with a certain cognitive status in the situation model of interlocutors (i.e., their mental representation of the discourse; Zwaan and Radvansky 1998). Based on factors in the linguistic discourse context, entities will vary in their **accessibility** (other terms are *salience* or *topicality*; Chafe 1996; Givón 1983; Gundel et al. 1993; Ariel 1990). Under these accounts, third-person pronouns (e.g., "she") are specialized for highly accessible entities, while specific expressions (e.g., "Laura Aina", "the PhD student from Pompeu Fabra University") are required when an entity is highly inaccessible. Factors modulating the accessibility of an entity have been identified by looking at contexts where pronouns or other reduced expressions tend to be used, both in corpora and elicitation experiments. For instance, pronouns are more likely to be used for entities that have been mentioned recently, or in subject position (Arnold, 1998; Stevenson et al., 1994).

Another notion that has been put forward to explain reference production is **referent predictability**, or *context informativeness*: if a context is informative about a referent, this entity becomes predictable, to some degree, without accessing the referring expression itself. These accounts explain reference production as a process aimed at making communication not only effective but also efficient. In general, speakers would try to be neither overinformative nor underinformative (Grice, 1975) and avoid redundancies in communication (Jaeger, 2010). During reference production, this would lead to taking into account both clarity (i.e., that their expression is understood by the interlocutor) and cost (i.e., that the expression is easier for them to produce; Tily and Piantadosi 2009; Orita et al. 2015). Since informative expressions are typically longer (e.g., "the student" vs. "her"), they would be used only when necessary to understand the utterance. On the contrary, if a referent is highly predictable from the context, it can be referred to with a reduced, less informative expression like a pronoun. An intermediate account is proposed by Arnold (2001), taking accessibility to be influenced by how likely a referent is expected to be mentioned again in the discourse.

Less informative expressions are more likely to cause ambiguities: as they provide little information about the referent, they may apply to more than one entity in the context. The predictability account suggests that speakers are prone to create ambiguities when the interlocutor has enough contextual information to individuate the referent (as in (15)). On the contrary, the ambiguity is avoided, by using a more explicit expression, if the context is not informative enough (as in (16)).

(15) Barbara gave Lisa a coffee since *she* was feeling sleepy.

(16) Barbara and Lisa were having a coffee, when *Lisa* got an unexpected call.

While this seems like an intuitive explanation, empirical evidence does not actually point to it consistently. Concretely, factors that modulate referent predictability do not seem to affect the choice of a referring expression, and in particular pronominalization, to the same degree. Several studies on this topic are based on psycholinguistic experiments, manipulating semantico-pragmatic factors that affect expectations about referents. Some researchers focused on the effect of *implicit causality* verbs (Fukumura and Van Gompel, 2010; Rohde and Kehler, 2014): a verb like “congratulate” favors the object to be mentioned next (Lisa in (17a)); by contrast, “surprise” favors the subject (Barbara). Other studies looked at *transfer of possession* verbs (Stevenson et al., 1994; Rosa and Arnold, 2017), which create a bias for the entity in the thematic role of goal (in (17b), Lisa with “give”, and Barbara with “received”). The interaction of verb semantics with coherence relations was also investigated (Stevenson et al., 1994; Fukumura and Van Gompel, 2010), in particular the effect of the connectives *because* and *so* (17c). These experiments typically employed discourse continuation tasks to probe the expected referents and the form used to refer to them (predictability and production, respectively). This is achieved by asking subjects to complete an unfolding discourse as in (17) before the pronoun. But also one can look at referent identification judgments (comprehension), by asking to continue the discourse after the pronoun (“she” in (17)).

(17) a) Barbara congratulated/surprised Lisa. (She) ...
b) Barbara gave Lisa a book/received a book from Lisa. (She) ...
c) Barbara congratulated/surprised Lisa because/so (she) ...

Some studies (e.g., Arnold 2001; Rosa and Arnold 2017) reported that semantico- pragmatic factors that affect the predictability of an entity also influence reference production (pronouns are used for predictable entities). However, other research (e.g., Stevenson et al. 1994; Fukumura and Van Gompel 2010; Rohde and Kehler 2014) found evidence that reference production is instead mainly driven by syntactic factors; in particular, a strong preference to use pronouns to refer to subjects.

The debate over the relation between predictability and referential form was addressed also considering corpus data and the elicitation of cloze task judgements (Tily and Piantadosi, 2009; Modi et al., 2017). Humans were instructed to read a passage of text and indicate the referent that they expected to be mentioned next, in order to estimate referent predictability. The role of this on reference production (its syntactic type: pronoun, description, or name) was then studied jointly with other discourse factors such as recency and syntactic prominence (subjecthood). Tily and Piantadosi (2009) considered data from news text and found that predictability affects reference production, including beyond the role of the aforementioned discourse factors. However, these factors still help to predict the type of a referring expression, pointing to a complex interaction with predictability. By contrast, Modi et al. (2017) focused on texts describing typical events (e.g., taking a bath, baking a cake, etc.) but did not find an effect of predictability on the choice of referring expressions. Considering existing empirical evidence, it is thus to date not fully clear whether predictability and an efficient approach to communication affect reference production strategies (Arnold and Zerkle, 2019).

2.2 Deep Learning and Linguistic Ambiguity

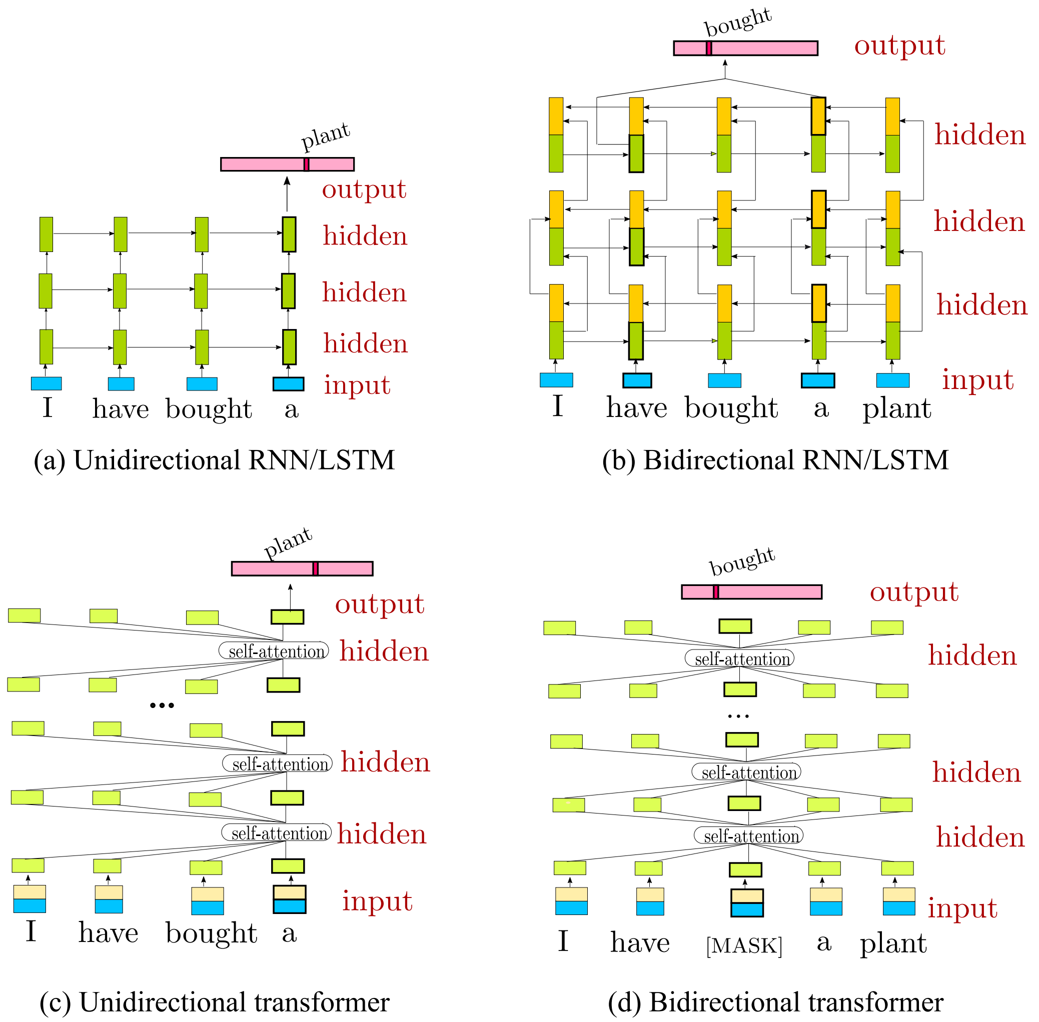
2.2.1 Deep Learning Models of Language

In the last decade, Deep Learning (LeCun et al., 2015; Schmidhuber, 2015) established as the main modeling paradigm in computational linguistics and NLP, leading to vast improvements in language technologies and providing new tools for the study of language (Linzen, 2019; Baroni, 2021). In this section, I provide a general introduction to this family of methods, as relevant to my thesis.

Deep learning is an instance of supervised machine learning. A model trained with supervision learns to solve a task from exposure to a set of data labeled for the correct prediction; i.e., the training data. For instance, the input may be a linguistic sequence and the output a word that follows it (e.g., "Today the supermarket is" → "closed"): the **language modeling** task. Another task of relevance to this thesis is **coreference resolution**: to group referring expressions on the basis of the entity they refer to (e.g., "Have you met the new neighbor? She's nice!" → ["you"], ["the new neighbor", "She"]). Supervised machine-learning methods other than deep learning tend to presuppose a considerable effort in selecting hand-crafted features to describe the input data; for instance, the syntactic role of noun phrases in predicting coreference relations. The computational model then learns what impact each feature has on the output predictions. Deep learning lifts the need for the preliminary feature engineering step, by learning its own representation of the data. This allows learning powerful computational models directly from raw data as input (in the case of language, from text or speech).

A deep learning model is a specific type of artificial **neural network** (Rosenblatt, 1958), where distributed (i.e., high-dimensional) representations of the input (also referred to as *states* or *activations*), are computed at multiple levels of abstraction: the *layers*. The initial and final layers are referred to as *input* and *output* layer, respectively, while the intermediate ones are the *hidden* layers. The number of hidden layers characterizes the *depth* of the neural network. Each representation is the (typically, non-linear) transformation of the representation at the previous layer, until a prediction is finally given at the output layer. This multi-layer representation learning is what makes deep learning models so powerful, in that it gives a model the capacity to autonomously discover the aspects of the data that are useful to succeed in the objective task.

In Figure 2.1, I schematically represent the architectures most commonly used to process text sequences focusing on the example of the language modeling task: *neural language models* (Bengio et al., 2003; Mikolov et al., 2010). For each architecture, I depict both the **unidirectional** and **bidirectional** version: the first uses only the previous context (left in English) for prediction, while in the second both sides are considered. In both architectures, a word token is represented at the input layer by means of an *embedding* that is specific to its word type. This vectorial representation remains constant



(a) Unidirectional RNN/LSTM

(b) Bidirectional RNN/LSTM

(c) Unidirectional transformer

(d) Bidirectional transformer

Figure 2.1: Architecture of RNN/LSTM and transformer language models in both their unidirectional and bidirectional format. The hidden and output states involved in the target predictions are marked with thicker borders.

across usages of the word.[4] A **Recurrent Neural Network (RNN**; Elman 1990) in its unidirectional version (Figure 2.1a), builds a representation relative to a certain token in the hidden layer as a function of 1) the current token representation from the previous layer, and 2) the hidden state from the previous token. These recurrent connections enable to retain information from the context while processing a new word; variants such as **Long Short-term Memory Networks (LSTMs**; Hochreiter and Schmidhuber 1997), used in my experiments, are designed to enable better retention of long-distance information. In a bidirectional RNN-based model (Figure 2.1b), the final prediction depends on two RNNs processing the text in opposite directions.

By contrast, a **transformer** architecture (Vaswani et al., 2017) uses self-attention to incorporate the context in the processing of the input: the representation of a token is a weighted average of that of context tokens from the previous layer of computation. The architecture is composed of a series of blocks, outputting each the result of multiple self-attention steps based on different transformations of the inputs (*heads*). To provide information about the order of tokens in a sequence, positional embeddings are given as part of the input. A transformer can be unidirectional (Figure 2.1c) if for each timestep it only computes self-attention using tokens in the previous context (Radford et al., 2018). The model is instead bidirectional if the full input sequence is accessed (Figure 2.1d). For language modeling, to cover the information from the input word for prediction this can be masked; i.e., substituted by a generic `[MASK]` token (Devlin et al., 2019).

In the latest years, the NLP community mostly focused on Transformer architectures, as they facilitate scaling the size of both the model and its training data, leading to more and more powerful computational models (for example, see the evolution of the GPT, GPT2, and GPT3 language models; Radford et al. 2018, 2019; Brown et al. 2020). Another important source of advances within the use of deep learning from NLP came from the practice of building computational models for a task by relying on pre-trained neural language models (Peters et al., 2018a; Ruder, 2019). The core idea behind this type of **transfer learning** is that language models, trained on a generic word prediction task on large unlabeled text corpora, learn to process an input sequence in a way that can be relevant also for other linguistic tasks (e.g., coreference resolution). This information can be transferred to another model by using the internal representations (hidden states) from the language model as input (instead of the static word embeddings). In this phase, the language model parameters may remain frozen, or get optimized (i.e., *fine-tuned*) to the target task (Peters et al., 2019). This methodology led to numerous advances in NLP tasks (e.g., Peters et al. 2018a; Devlin et al. 2019) and became the current mainstream pipeline in model building.

[4] Exceptions are models that use character or subword embeddings. In this case, the representation that remains static is that of a character or subword. I do not consider exceptions models that rely on transfer learning from language models (see Section 2.2.1), and thus use "contextualized embeddings" as input. As these come from the hidden states of a language model, still the starting point of the processing of the sequence is at the input layer of the language model, which involves some static embeddings.

The success of deep learning methods in NLP led to the emergence of a thriving research area focusing on the internal mechanisms underlying the behavior of deep learning models (Alishahi et al., 2019; Belinkov and Glass, 2019). This research arises both from a need to enhance the interpretability of these models, as well as a linguistic interest in the learning capabilities of data-driven models (in particular, language models that are only exposed to unlabeled corpus data). The internal representations of deep learning models, as they are, are not directly telling of the information encoded nor the dynamics causing their behavior. Also, overall performances on generic benchmarks do not necessarily clarify the type of linguistic knowledge a model acquired. This led to the development of **analysis techniques** aimed at uncovering the internal dynamics of such "black-box" models and assessing their ability to account for specific phenomena.

For the purpose of this thesis, we can distinguish two main families of analysis approaches. First, one can clarify the workings of deep learning models by looking at the information encoded in the **internal representations**. A way of probing this is by means of *auxiliary* supervised tasks (alternative terms are *diagnostic* or *probing* tasks; Adi et al. 2017; Conneau et al. 2018; Hupkes et al. 2018): One trains a separate, simple machine-learning model on some task (for instance, part-of-speech tagging) using as input the states of interest. If successful, we can conclude that the representations contained information relevant to the task considered. Another way of studying the information in hidden states is by instead looking at the way these representations organize in the high-dimensional space (e.g., Saphra and Lopez 2019; Ethayarajh 2019). As an alternative analysis method, one can focus on the **behavior** (i.e., predictions) of a model on data focusing on a specific phenomenon. This can clarify the knowledge that is taken into account by the model in its predictions by simply examining its output on these targeted data. An example of this type of evaluation is the *subject-verb agreement* task (Linzen et al., 2016; Gulordava et al., 2018b), where one assesses whether a higher word probability is assigned to the verb form that agrees in number with the subject, or to the incorrect alternative (e.g., "The windows of the building *are* vs. *is*"). This allows establishing if the model keeps track of syntactic structure. Other works focused instead on semantico-pragmatic aspects: for instance, Ettinger (2020) and Pedinotti et al. (2021) looked into the ability of language models to take into account information such as commonsense reasoning and event knowledge. In my experiments, I will use analysis techniques both focusing on the information encoded in the internal representations of a model and on its output behavior.

2.2.2 Ambiguity Resolution in Deep Learning Models

In this subsection, I provide an overview of what we know about the resolution of linguistic ambiguities in deep learning models, on the basis of both their inner structure and previous work. To guide this discussion I will follow the computational graph of a deep neural network, sequentially focus on the input, hidden and output layers (from the input data to the output predictions; Figure 2.1).

Before moving on to this, let me first introduce an important distinction, which can have an impact on both the information encoded by a model and on ways in which we can inspect it. A deep learning model is explicitly trained to resolve ambiguities when the output of the task corresponds to the analysis of a linguistic sequence. To give an example, a syntactic parser will derive a representation of the sentence structure (e.g., dependency graph): its predictions will thus reflect the resolution of syntactic ambiguities. Analogously, a coreference system will establish chains of mentions referring to the same entity, thus reporting the predicted referent of ambiguous expressions. Many of the classic NLP tasks can in fact be considered to involve some kind of ambiguity resolution (Manning and Schutze, 1999). On the contrary, some models carry out more "generic" tasks, where interpretations of expressions are not explicitly output, though they may be implicit in the behavior. Two prominent examples are the tasks of language modeling and machine translation. Given a processed sequence, the model is trained to predict a word or a translation equivalent in the target language. The resolution of ambiguities might affect the predictions: e.g., "palm" should be translated in Italian as "palma" when meaning 🌴 and as "palmo" when meaning ✋. However, the model does not get any direct supervision about ambiguity resolution. Nevertheless, it can learn to implicitly infer both interpretations of expressions and their effect on the task, as part of training.

Input layer: Word embeddings

In the standard case, the occurrence of a certain word in an input linguistic sequence is represented by means of a **word embedding**.[5] This word representation is obtained by optimizing performances in the objective task on training data, consequently abstracting over a diversity of occurrences of the word. It thus reflects the information that a word is generally associated with, but, at this stage, does not vary depending on the context.

The representational format of word embeddings has much in common with **distributional semantics** (Lenci, 2008; Boleda, 2020). Within this framework, words are represented with fixed-size high-dimensional vectors based on their distribution across contexts in a text corpus. This is based on the distributional hypothesis (Harris, 1954; Firth, 1957): the more aligned is the usage of two words, the more similar is the content they express. One can then encapsulate the distribution of a word in a vectorial representation, and, comparing it to that of other words, establish similarity, or relatedness, in content. In particular, this can be computed in terms of geometric distance (e.g., cosine) between vectors. As this notion of similarity is a continuous matter, differences among words are accounted for in a **graded** way. Neural network models can be used to derive distributional representations (Mikolov et al., 2013), though the network is typically not

[5] See Footnote 4 for exceptions.

deep (i.e., only one hidden layer).

Word embeddings from deep learning models, however, have very similar properties to distributional representations, and, at least in some cases, qualify as such. In particular, the embeddings from neural language models, as they are learned to predict a word based on its context, are essentially optimized to reflect distributional patterns.[6] Like distributional semantic representations, word embeddings from deep learning models have the following properties that are relevant to ambiguity.

- They are **static** word representations; i.e., they do not change across contexts (except as they are optimized during training). They represent information that is relevant to the **lexical** – generic and out-of-context – content of a word (Carston, 2012), but cannot, by design, account for the variation in meaning across usages (Erk, 2010; Westera and Boleda, 2019). They can be seen as the starting point for the process leading to infer the intended interpretation of a word, but it is not at the input layer yet where lexical disambiguation occurs.

- Due to being static but optimized across contexts, these word representations will reflect the diversity of usages, and therefore ambiguity, of a word. If a word appears in the training corpus with multiple interpretations (e.g., "mouse" as 🐭 or 🖱️) and these distinctions are relevant to the task at hand (e.g., word prediction), this diversity will affect the learned representation. As a result, word embeddings tend to encode multi-faceted information relevant to different interpretations of the word (Camacho-Collados and Pilehvar, 2018). Within this thesis, I refer to this property as **underspecification**, following the terminology adopted in linguistics (Frisson, 2009; Falkum and Vicente, 2015). Senses are not explicitly encoded nor separated in a word embedding (they are "merged" in a unique vector), but the position in the high-dimensional space can reveal their effects (e.g., the embedding of "mouse" is close to both "rat" and "cursor").

- The frequency of a word sense in the training corpus affects the extent to which that interpretation is represented in a word's vector: the lexical representation will reflect that type of usage the more this is dominant in the data (Arora et al., 2018). That is, distributional representations are, with respect to senses, **frequency-biased**. We can thus consider that these word representations encode sense dominance: information about frequent senses will be more activated by default. This aspect has some parallel in human processing (Duffy et al., 1988).

[6] The difference with traditional distributional vectors is that these are learned predicting simple co-occurrences in a window of words, disregarding word order in a sequence, whereas language models take into account the word's position.

Hidden layers

Independently of the architecture (e.g., RNN/LSTM, transformer), when a word embedding is passed as input to a model, this is combined in the hidden states with information coming from the other words in the context (Figure 2.1). Hidden states will eventually affect the output computed by the model; they are therefore optimized to represent information that is useful in order to succeed in the objective task. However, as mentioned earlier, it is not a priori known what this information is, as the model autonomously learns the aspects of the data to encode and then leverage for its predictions.

In models that resolve ambiguities at the output, we can expect that in the hidden layers the model attends to information that tears apart the various interpretations of a word. The model will have to represent occurrences of expressions associated with different interpretations sufficiently differently in order to enable a successful output classification. For instance, Aina et al. (2019) provides a visualization of this idea in the context of a referential task (to classify mentions by the entity they refer to): We show that hidden states relative to mentions form clusters in the high-dimensional space based on the entity they refer to. The clusters get further away from each other as the output layer is approached.

In terms of the type of information encoded in the hidden states, much focus has been put on the analyses of models trained on generic tasks like language modeling. Various works reported evidence that after processing a linguistic sequence such hidden states encode much and diverse information (Adi et al., 2017; Conneau et al., 2018; Tenney et al., 2019b). This includes both superficial information, like the identity of the processed words, and information resulting from deeper linguistic processing, like that relevant to syntactic and semantic parsing. Moreover, using hidden states from language models as contextualized embeddings to pass as input to a model tends to lead to substantial improvements across tasks (Peters et al., 2018a; Devlin et al., 2019). This supports the idea that these representations encode information that is of sufficient quality, and at the same time general enough, to facilitate various NLP tasks.

The information in the hidden states arises by means of a process of *contextualization* of the lexical information passed as input (the word embedding). This contextualization may encompass word-level disambiguation: to identify the contextual content of the word given as current input (e.g., its part of speech, sense, or referent). But also composition: putting together the contribution of different words in a sequence to then represent the contextual analysis of a larger expression. Previous research provided evidence that hidden states reflect the context-dependent content of both words and larger expressions, at least to some extent. In the following, I will summarize the main findings, focusing on the types of ambiguity I study in this thesis.

When it comes to lexical ambiguities, studies focused on whether hidden states from language models can distinguish occurrences of different senses (Pilehvar and Camacho-Collados, 2019; Reif et al., 2019; Wiedemann et al., 2019), as well as accounting for graded differences across usages (Garí Soler et al., 2019a; Nair et al., 2020; Trott and Bergen, 2021). Other works focused on the effects of composition on word meaning (Shwartz and Dagan, 2019; Yu and Ettinger, 2020). Overall, there is evidence that, in spite of some margin for improvement, the hidden states from language models encode context-sensitive word information and account well for the variation of a word's meaning across contexts. Consequently, these representations have been used as proxies for a word's contextual meaning (Li and Joanisse, 2021) or the degree of lexical ambiguity (Pimentel et al., 2020a) for the purpose of linguistic studies. Regarding the resolution of referential ambiguities, a few works provided evidence that the information in hidden states is predictive of anaphoric relations of pronouns (Peters et al., 2018b; Jumelet et al., 2019; Sorodoc et al., 2020; Davis and van Schijndel, 2020). Current evidence indicates that language models track well grammatical information involved in coreference (e.g., gender agreement), and also, to some degree, distinguish mentions of different entities. Finally, much work focused on the encoding of syntactic analyses in language models' internal representations. It was shown that the hidden states can be predictive of sentence structure information such as the subject's number (Giulianelli et al., 2018), the maximum depth of a parse tree (Conneau et al., 2018), the part of speech of a word (Tenney et al., 2019a) or the distance in the parse between two words (Hewitt and Manning, 2019). However, these works do not specifically focus on cases where multiple syntactic parses are either locally or globally possible, and therefore how a language model resolves ambiguities.

Output layer

When reaching the output layer, a deep learning model typically outputs a probability distribution over some classes, based on which a prediction is made. As I anticipated earlier, such prediction can itself directly reveal the resolution of an ambiguity. It is then possible to assess whether the model took the correct interpretation by directly inspecting the output on the task. For instance, if a part-of-speech tagger processes a sentence containing the word "show", we can see if this was labeled as noun or verb. Deep learning led to numerous advances across NLP tasks, thus including improvements in predicting correct interpretations of expressions. Transfer learning from language models led to even more progress on top of that. To give an example, Lee et al. (2017) introduced a successful deep learning architecture for coreference resolution, which, unlike previous models, did not require a preliminary feature engineering step. Passing representations from language models as input to an architecture of this kind led to further advantages (Lee et al., 2018; Joshi et al., 2019, 2020). Nevertheless, ambiguities can still pose some challenges to NLP systems, as shown by evaluations on challenge sets – i.e., data focusing on harder instances of a task. Some examples are the dataset by Elkahky et al. (2018) on part-of-speech-tagging of lexically ambiguous words (Noun/Verb), or that by Webster et al. (2018) on the coreference resolution of ambiguous pronouns.

When we focus on the output of models trained on generic tasks such as language modeling, it is instead more difficult to think of ways in which we can evaluate their ambiguity resolution skills. A word prediction may depend on the resolution of an ambiguity: e.g., After processing "The TV shows", the probability of "are" will be affected depending on whether "shows" is interpreted as a verb or a noun. However, given the countless ways disambiguation could occur in terms of word predictions, it is not easy to transfer this intuition in a concrete task to systematically evaluate language models.

Still, a few works proposed analyses of language models that are relevant to ambiguity resolution by inspecting its output predictions, besides its internal representations. An example is the deployment of language models on sentences with temporary syntactic ambiguities (e.g., the example of (13)), and, akin to studies on human processing, looking for evidence of garden-path effects (Futrell et al., 2018, 2019). In particular, a high *surprisal* (i.e., negative log-probability) at the disambiguating point suggests that the model preferred the alternative interpretation of the input. This was interpreted as evidence that language models maintain an internal representation of the syntactic state of an unfolding sentence, which is incrementally updated. Another line of work tested whether language models exhibit human-like referential expectations (Davis and van Schijndel, 2020; Upadhye et al., 2020), modulated by verb semantics (e.g., implicit causality verbs) and rhetorical relations. These works compared the probability of the pronouns "he" and "she" in contexts like (17) but with entities of different gender, allowing to establish which entity the model expects to be mentioned. This is not directly an evaluation of ambiguity resolution, as it does not test how the model interpreted a referring expression. It however assesses whether the language model is sensitive to discourse factors that can impact ambiguity resolution.

Overall, both the architecture of deep learning models and previous analyses suggest that these have some degree of the capacity to handle ambiguities: in particular, to encode and use both the expression-specific and the context-sensitive information that is necessary to infer the correct readings. Nevertheless, there are open questions about their degree of success and underlying dynamics in resolving ambiguities. Chapters 3 and 5 study the way neural language models process lexical and syntactic ambiguities, studying, in one case, the information encoded in their hidden states and, in the other, their output behavior. But there are other ways in which deep learning models can be relevant to the study of ambiguity, besides uncovering their way of dealing with the phenomenon. Accounts in linguistics and cognitive science posit comprehension and production strategies connected to ambiguity. Testing these hypotheses on a large scale can be facilitated by the use of reliable computational estimates, substituting the elicitation of human judgments. Deep learning can help us in this, by computing scores or high-dimensional representations that can act as a proxy for, e.g., contextual expectations, or the lexicon. This is the approach I take in the chapters 4 and 6.

Chapter 3

LEXICAL AMBIGUITY RESOLUTION IN NEURAL LANGUAGE MODELS

In this chapter, I investigate how lexical ambiguities are resolved in neural language models. Lexical ambiguity is broadly defined as the phenomenon where a word type is associated with multiple readings (Cruse, 1986; Small et al., 2013; Rodd, 2018). To exemplify this multi-faceted phenomenon, we can consider the highly ambiguous word "match" in English, which is both syntactically and semantically ambiguous, and encompasses senses with more or less conceptual overlap:

(18) a) Those trousers *match* your shirt. 👌 Verb
b) Avocado and eggs are a great *match*. 👌 Noun
c) They stopped dating because they were not a good *match*. 💞 Noun
d) The national anthem was played before the football *match*. 🏟️ Noun
e) We need a *match* to light the candle. 🔥 Noun

Lexical ambiguity affects virtually all words in a lexicon, at least to some degree (Cruse, 1986). Yet, in the vast majority of cases, it does not impair human communication: the correct interpretation can be smoothly identified in context. An artificial system processing language also needs to deal with lexical ambiguities: they are pervasive, and their resolution has consequences on both the syntactic structure and meaning of a text (as shown by the different contributions of "match" in (18)).

In this chapter, I study how deep learning models process lexical ambiguities. In particular, I develop a methodology to quantify the retention of underspecified lexical information in a model's internal activations, contrasted with the representation of the contextually relevant interpretation. This analysis of deep learning models is related to psycholinguistic research on the activation of information relevant to different senses during human processing (*word meaning access*) and its relation to ambiguity resolution (Frisson, 2009; Vitello and Rodd, 2015).

I focus the analysis on neural **language models** (LMs), deep learning models that are trained to predict the word that fills a certain slot in a linguistic sequence (e.g., in (18a), to predict “shirt” based on “Those trousers match your...”). Because of their generic objective, LMs allow to observe a strategy to the resolution of lexical ambiguity which emerges only from exposure to word usages in a corpus, without explicit signal nor built-in mechanisms to handle the process. This offers a comparison to human behavior and insights into the type of mechanisms that make word interpretation possible. Furthermore, the study informs us about the context-sensitivity of LMs to the variation of a word’s meaning, and thus their ability to successfully cope with ambiguities. Methodologically, the study reported in this chapter fits into the research branch analyzing the behavior and underlying dynamics of neural network models of language (Belinkov and Glass, 2019), in particular using supervised auxiliary tasks.

3.1 Approach and Research Questions

In the last years, LMs have received much attention in NLP due to the advantages they offer when used for transfer learning to other tasks (Peters et al., 2018a, 2019; Devlin et al., 2019). The improvements across tasks that this technique introduces are often traced back to the context-sensitivity of the hidden states of LMs, which can be used as *contextualized* embeddings of the input words (e.g., Peters et al. 2018a) – as opposed to the use of static word embeddings (Mikolov et al., 2013). But what exactly do these representations encode? Indeed, another reason why LMs have been much in the spotlight is to uncover the type of linguistic knowledge that these models acquire. On the one hand, the information that LMs encode can justify the advantages of their use for transfer learning. On the other, as LMs are trained simply from unlabeled text corpora, we can observe what kind of linguistic knowledge is autonomously acquired without the need for explicit supervision, and what processing strategies this gives rise to. For instance, it was shown that LMs track syntax to a large extent (see Linzen and Baroni (2021) for an overview) but also aspects of semantics and pragmatics (e.g., Shwartz and Dagan 2019; Ettinger 2020).

Precisely because of the claim that hidden states from LMs offer contextualized word representations, various works looked into the extent to which they account for the contextual interpretation of a word (see Loureiro et al. 2021 for an overview). This was studied both at the level of distinguishing occurrences of different senses (e.g., Pilehvar and Camacho-Collados 2019; Reif et al. 2019) and accounting for graded differences among word usages (e.g., Ethayarajh 2019; Garí Soler et al. 2019b). Overall, these works show that the variation between a word’s interpretation across contexts correlates with the geometric distance of the hidden states in the high-dimensional space: the more distant the content conveyed in different occurrences, the further the hidden states tend to be. But how is this contextualization achieved by the LM? And what type of word information is encoded in the hidden states? This chapter aims to shed light on these aspects.

As described in Section 2.2.2, in a LM the processing of a word starts with being inputted a static fixed-sized vector: the **word embedding**. This representation is learned during training while optimizing word predictions across the training corpus. We can think of the word embedding as the word information initially activated by a LM during processing. Akin to a distributional semantic representation (Lenci, 2008; Boleda, 2020), the word embedding represents the **lexical information** about the word that the model has acquired: This is static – it does not change across contexts (after training) – and underspecified – it represents potentially multi-faceted content associated with the word (e.g., “match” as 🔥 or 🏟️), with a bias towards the most frequently attested senses. The position in the high-dimensional space of the word embedding can reveal the effects of the different senses: e.g., the embedding of “match” can be close to both “fire” and “sport”. The idea that, during interpretation, underspecified and frequency-biased information is initially activated has, with some degree of simplification, a parallel with what is found in the case of humans (Duffy et al., 1988; Rayner and Frazier, 1989).

In the hidden states, this lexical information is “put in context”: the word embedding is combined with representations from other tokens. Though neural language models are not explicitly trained to resolve ambiguities, detecting the context-specific contribution of a word may aid the objective task (word prediction). It is therefore plausible that the LM learns to resolve the ambiguity in the input word embedding, so that the output predictions can depend on the correct interpretation of the word. To achieve this, the model would need to infer and encode the information that is relevant to the correct interpretation in its internal representations (the hidden states), jointly with other information deemed useful for the output prediction. Borrowing a term from lexical semantics, this ambiguity resolution process in the hidden layers can be seen as one of **modulation** (Cruse, 1986): The multi-faceted content in a lexical entry – in this case, of a word embedding – is adjusted to circumscribe the contextually-relevant contribution of a word.[1] In this chapter, I investigate the nature of this modulation process in LMs: the extent to which it leads to encoding contextual word information, and whether this occurs with some degree of retention of the underspecified lexical information that was initially passed as input.

We assess the presence of lexical and contextual word information not only in hidden states that process a word as input – those that have to resolve the lexical ambiguity – but also, as a contrast, those that instead need to predict the word at the output (without

[1] In Lexical Semantics, *meaning modulation* is typically used to refer to polysemy resolution, and therefore distinguishing related senses of a word. By contrast, we here collapse all cases of ambiguity resolution, including over unrelated senses (e.g., homonymy) into the same mechanism.

having observed it; e.g., in (18) predicting a word to fill the slot of “match” based on “Avocado and eggs are a great...”). To succeed at the word prediction, the model needs to form expectations about the information that is more likely to be expressed: While in (18e) it should expect things you can use to light a candle (“match” included), it should not do so in (18b). We analyze the contextual expectations formed in hidden states of LMs, in order to isolate and evaluate the contribution of the context alone (without access to the lexical information as input).

For this analysis, we use supervised auxiliary tasks (Adi et al., 2017; Conneau et al., 2018; Hupkes et al., 2018) to probe the information encoded in the internal activations of a model. These methods are used under the assumption that if some information is present in the representation of a deep learning model (e.g., in a hidden state) then a (simple) *diagnostic* model should learn to extract this information from the analyzed representation (e.g., if a word representation distinguishes the part-of-speech, then one can predict it from this representation). A classification task is typically used as diagnostics, while we instead focus on a representation learning objective. We train a mapping from internal representations from LMs to word representations as a way to extract – and then evaluate – the word information encoded in the hidden states of a LM.

With this methodology, we address the following research questions about how lexical ambiguities are processed across the hidden layers of a LM:

RQ1. When a word is inputted for processing to a LM, to which extent is the lexical information in the word embedding retained across hidden layers?

To resolve the ambiguity of a word, the LM should get to represent information that is relevant to the contextually appropriate interpretation in its hidden layers. This, however, does not necessarily exclude that some degree of retention of the original inputted lexical information could occur. In humans, for instance, there is evidence that underspecified information relevant to different senses of a word is initially activated during word meaning access, though not maintained long once an interpretation is selected (Vitello and Rodd, 2015). In the case of a LM, I consider different two scenarios to be possible: Interpreting a word may lead to suppressing (deactivating) the information that is not relevant to the selected interpretation, and only what is relevant is passed to the next steps of processing (i.e., layers). Alternatively, the LM may have a way of selecting and representing the interpretation of a word that however proceeds independently from some degree of retention of the inputted lexical information.

RQ2. How good is the representation of contextual word information in the LM? What differences do we find across hidden layers?

The amount of information relevant to the contextual interpretation of a word we would find in the hidden layers may depend on several factors. First and foremost, it will be a signal of the level of context-sensitivity of the LMs, which we aim to assess. We can expect models that are bigger and trained on more data to be superior in this. Moving across hidden layers (from the input to output), we could find a better or worse reflection of contextual word information. This may depend on the architecture and the way output predictions are computed in the LM. For instance, a deep model may exhibit a progressive contextualization of word information, distributing and improving the process across layers moving towards the output. But also, if at the output a word other than the input (e.g., the next word) is predicted, it is possible that information about the input word actually decays towards the output to focus instead on the word to predict (see Voita et al. 2019 for an analogous argument).

RQ3. How are the lexical and contextual word information represented in a hidden layer that, instead of processing a word, predicts it at the output?

Hidden states used to predict a word at the output do not observe the target word but only its linguistic context. Therefore, looking at the word information in these states means looking into what can be determined based on context only, without access to lexical information; else put, the contextual expectations of the model relative to that slot in the text. We expect these hidden states to contain word information too, though of lower quality to information obtained after processing the word, due to the uncertainty over the word to predict. As this word information is derived from context only, it will be context-sensitive by design, and, in contrast to states that process the word as input, should not exhibit any trace of the underspecification in the word embedding. Again, better LMs are likely to predict better word information, as this depends on the ability to predict words in the first place (the objective task).

3.2 Hidden States from Language Models

3.2.1 Language Models

In this chapter, we study the behavior of two LMs. Both are bidirectional, in that a word is predicted based on its surrounding context (previous and following). This choice is motivated by the following reasons: First, the data which we use for the analyses were obtained through annotations where subjects could access the surrounding context of a word. We thus give a LM the same amount of context that the humans received for the task. Second, though humans process linguistic sequences incrementally, cues to a word's interpretation may appear both in the previous and following context of a word (Foraker and Murphy, 2012). We focus on the representations obtained once all information available for disambiguation is given, leaving aside the incrementality of the interpretation process. We however look at this more directly in Chapter 5, when focusing on the processing of syntactic ambiguities with unidirectional LMs. Though the two LMs analyzed share the property of being bidirectional, they differ across a notable set of dimensions.

biLSTM. This LM consists of a bidirectional LSTM (Schuster and Paliwal, 1997; Hochreiter and Schmidhuber, 1997) with 3 hidden layers. The architecture is schematically represented in Figure 3.1a. Each input word at timestep t is represented through its word embedding (size 300); this is fed to both a forward and a backward 3-layers LSTMs (hidden size 600/600/300), which process the sequence left-to-right and right-to-left, respectively. To predict the word at t, we obtain output weights by summing the last hidden states of the forward and backward LSTMs at timesteps $t-1$ and $t+1$, respectively, and applying a linear transformation followed by softmax (Eq. 3.1, where L is the number of hidden layers). This ensures that a word is predicted jointly using both its left and right context, but without access to the word itself. In this sense, the architecture resembles, for instance, that of the context2vec model by Melamud et al. (2016) but not the biLSTM used for ELMo (Peters et al., 2018a), instead optimized to predict the word from the two sides of the context separately.

$$P(\mathbf{y}_t) = \text{softmax}(W^o\,(\overrightarrow{\mathbf{h}}^L_{t-1} + \overleftarrow{\mathbf{h}}^L_{t+1})) \tag{3.1}$$

The biLSTM was trained on a 150M tokens English corpus, consisting of the concatenation of English text data from a Wikipedia dump[2], the British National Corpus (Leech, 1992), and the UkWaC corpus (Ferraresi et al., 2008). The model has a vocabulary of 50K word types. More details about the training of the biLSTM language model are specified in Appendix A.1.

BERT. BERT (Devlin et al., 2019) is a transformer (Vaswani et al., 2017) encoder, using bidirectional self-attention (Figure 3.1b) It is trained using a masked language modeling objective:[3] At training, a sample of the input tokens is chosen to be predicted at the output. The input token is 1) 80% of the time substituted by the special token `[MASK]`, 2) 10% of the time substituted by a random token, 3) 10% of the time left unchanged. The identity of the token is then predicted at the output in correspondence to its position at the input; such prediction can be computed both when the token is masked, and – more easily – when it is given. The vector fed as input to the first hidden layer is the sum of the input token embedding with position and segment embeddings; we refer to this as $\mathbf{h}^0$, as it corresponds to a first layer of processing of the token embedding.[4] This representation is then passed through a set of hidden layers (transformer blocks), where

[2] From 2018/01/03, `https://dumps.wikimedia.org/enwiki/`

[3] A next-sentence prediction objective is also employed: given two chunks of texts as input, BERT has to predict whether one is the continuation is the other in correspondence of a special token `[CLS]`. This is to enhance the quality of transfer learning to text classification tasks. We do not focus on this objective as not directly relevant to our analyses (we deploy the model on a single chunk of text).

[4] Position embeddings are passed in order to inform the model about the token's position in the sequence (Vaswani et al., 2017). Segment embeddings are used to signal that a token belongs to a certain chunk of text: this plays a role during training for the Next Sentence Objective, but not in the way we deploy the model, where all tokens belong to the same chunk.

bidirectional self-attention (attending over other tokens, both at the left and right of the current one) is computed multiple times (number of *heads*); the resulting representations are concatenated and passed through a feedforward network to compute the output of the layer. The probability scores relative to the word at position t are computed, given the last hidden state $\mathbf{h}_t^L$, by first applying a non-linear transformation, then a linear one whose weight matrix is the transpose of the input word matrix (W_i^T; i.e., the weights of the input and output matrices are shared or *tied*; Inan et al. 2017; Press and Wolf 2017).

$$\mathbf{u}_t = \text{geLU}(\mathbf{h}_t^L W_u + \mathbf{b}_u) \tag{3.2}$$

$$\mathbf{o}_t = \text{softmax}(\mathbf{u}_t W_i^T + \mathbf{b}_o) \tag{3.3}$$

We employ the case-preserving version of BERT-base,[5] trained on a 3.3B tokens English corpus, made of the BooksCorpus and English Wikipedia. The model has 12 hidden layers with 12 attention heads, and both the hidden and embedding sizes are set to 768 units. BERT employs WordPiece token embeddings (Wu et al., 2016) with a vocabulary of 30K types comprising both full words and subwords. This implies that certain word type may not be represented through a word embedding, but split into more than one subword components, each processed as a separate token with their respective embeddings (e.g., "playful" → "play" + "##ful"). For simplicity, we restrict our analysis to word types that are not split into subwords in BERT, and are thus represented by a unique token embedding.

3.2.2 Processing and Predictive Hidden States

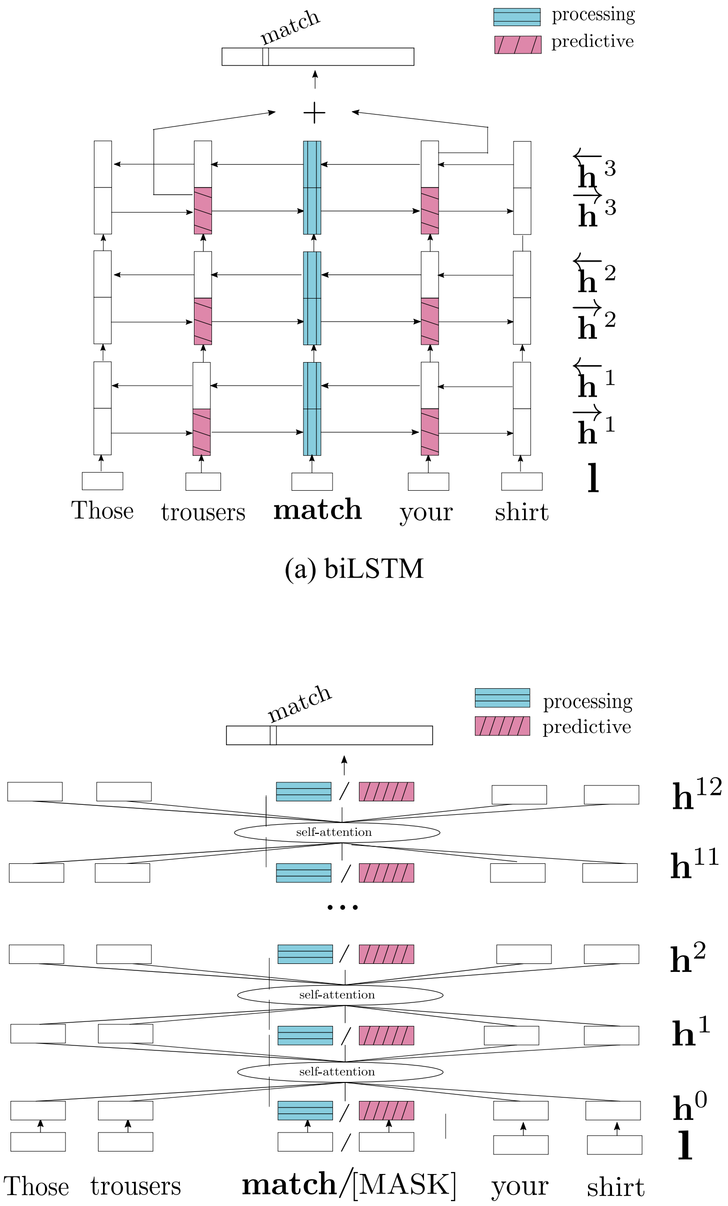
We analyze the internal activations in the hidden layers of a LM resulting from being passed a certain word as input. We refer to these as **processing states**, as they reflect how the LM has processed the target word. We contrast these representations with the internal activations used to instead predict a word as output – the **predictive states**. These are obtained without access to the target word but can be expected to reflect the information accumulated about it to succeed in the prediction.

The way these states are obtained depends on the architecture of the model, and the way word predictions are computed. This is visually represented in Figure 3.1. In the biLSTM, relative to a word at position t and hidden layer i, we consider as processing state the sum of the outputs of layer i from each unidirectional LSTM at t. For predictive states, we instead consider the states at the adjacent positions.

$$\mathbf{h}_{\text{proc}}^i := \overrightarrow{\mathbf{h}}_t^i + \overleftarrow{\mathbf{h}}_t^i \tag{3.4}$$

$$\mathbf{h}_{\text{pred}}^i := \overrightarrow{\mathbf{h}}_{t-1}^i + \overleftarrow{\mathbf{h}}_{t+1}^i \tag{3.5}$$

[5] Through the Transformers library.



(a) biLSTM

(b) BERT

Figure 3.1: Schematic representation of the architecture of the biLSTM and BERT language models, highlighting processing and predictive hidden states.

In BERT, both processing and predictive states are the output of the hidden layer i relative to the position of the target word. What changes between the way the two types of states are obtained is only the input: For processing states the word is passed as input, therefore processing its embedding. For predictive states, the word is substituted by the `[MASK]` token in order to conceal its identity.

3.3 Methods

To answer our research questions, we require a methodology to inspect the word information present in a hidden state. We contrast two types of word information:

- **Lexical information**: generic (out-of-context) information associated with a word. For the LMs, this is encoded in the word embedding. We can think of this information as the interpretational space of the word that the LM initially considers, as a word embedding tends to reflect multiple senses (e.g., for “match” it may be close to “game” and “fire”).

- **Contextual information**: the contextual interpretation of a word (output of a successful ambiguity resolution). For the purpose of this study, we think of this as a vector in the same space of the word embeddings. Contrarily to lexical information (the word embedding), contextual information should only reflect the correct interpretation and not multiple senses at once (e.g., for “match” as 🏟️ it is close to “game” but not to “fire”).

We aim to quantify the lexical and contextual information in processing and predictive states. We cast this as the probing tasks of extracting and evaluating representations of the lexical and contextual information, respectively, from hidden states. To do so, we learn transformations – which we refer to as **diagnostic models** (Hupkes et al., 2018)– that extract one or the other from hidden states. We then take the resulting representations to be telling of the word information encoded in the hidden states. We talk about “extraction” of word information from hidden states as these contain more information than that relative to the word to process or to predict; for instance, about the sentence structure (Hewitt and Manning, 2019; Tenney et al., 2019a) or other words in the context (Klafka and Ettinger, 2020). Through the diagnostic models, we individuate the information of interest and map it to the space of word embeddings.

Looking at processing states can shed light on ambiguity resolution in LMs. First, the amount of contextual information will inform us about the context-sensitivity of the model and its changes across layers (RQ2). Then, the comparison between lexical and contextual information will clarify the relationship between ambiguity resolution and retention of lexical information (RQ1). More contextual than lexical information would indicate that the underspecified information from the input word embedding was “reduced” in the hidden layers to the amount that is relevant to the context. The opposite case would instead suggest that the retention of lexical information proceeds independently of the contextualization process. Predictive states should contain generally less word information than processing states, due to the uncertainty over the target word. Yet, as they predict a word based on its context, they should accumulate some contextually relevant word information approaching the output (RQ3).

3.3.1 Representing Lexical and Contextual Information

While we have representations of the lexical information the LM considers (the word embedding), we do not have ground-truth representations of the contextual information expressed by a word (i.e., its interpretation). We solve this by using **lexical substitution** annotations (McCarthy and Navigli 2009; henceforth, referred to as LexSub) as a way to build proxies of the interpretation of a word. LexSub annotations are a set of one-word paraphrases relative to a target word occurrence in a context (see Table 3.1 for examples). The substitutes reflect how ambiguities have been resolved by humans, including both graded meaning nuances and distinctions among syntactic categories. These interpretation judgments can be collected without assuming a pre-defined set of sense labels as done, for instance, in the case of word sense disambiguation (Navigli, 2009). This is an appealing aspect considering the challenge of determining a sense ontology that is satisfying both in terms of both coverage and level of granularity (Kremer et al., 2014). Concretely, for our experiments, we use the ConceptInContext (CoInCo) corpus by Kremer et al., 2014 where occurrences of words in context (max. 3 sentences) have been annotated with substitutes by multiple raters. Substitutes are provided for 15.6K word occurrences (7.1K nouns, 4.6K verbs, 2.5K adjectives, 1.4K adverbs) from two domains of the MASC corpus (Ide et al., 2010), namely news and fiction texts.

LexSub data are often used as evaluation benchmark for contextual representation of words, in tasks like substitutes retrieval, or ranking of candidates (e.g., McCarthy and Navigli 2007; Melamud et al. 2016; Garí Soler et al. 2019b). One aims to a representation whose similarity to the embeddings of other words reflect the paraphrases; that is, it is the closest to words that could express the same content. As mentioned earlier, to be close to words related to the relevant interpretation is a desideratum for a representation of contextual word information. We thus build a proxy for contextual information using annotations of substitutes, enforcing that they satisfy this property.

To be precise, we represent the lexical information of a word l through its embedding, referred to as $\mathbf{l}$. For contextual word information, we define an operation that “merges” the content of the word embedding with that of the given substitutes. This is inspired by previous work that proposed simple vector operations to combine word representations (Mitchell and Lapata, 2010; Thater et al., 2011, a.o.). We represent the contextual information of a word **c** as follows:

$$\mathbf{c} := \frac{\sum_{i \in S \cup \{l\}} \mathbf{i}\, n_i + \alpha}{\sum_{i \in S \cup \{l\}} n_i + \alpha} \tag{3.6}$$

S is the set of substitutes of the word, **i** is the word embedding of the substitute i, α is a constant smoothing factor (set to 2), and n_i is for a substitute the number of annotators that produced it, and for the target word the mean of these values. We set α based on the parameter that would create representations that better retrieve and rank substitutes based on evaluation of validation data (see next section for the evaluation scores).[6] In intuitive terms, what this averaging procedure does is to highlight the information shared by the word and its substitutes at the expense of word information that is lexically but not contextually relevant. This is operationalized by increasing or decreasing values with respect to certain dimensions of the word vectors. We focus on the common core among these words, pointing to what makes them good substitutes in this context.

As substitutes will depend on the interpretation of the word, the resulting representation of the contextual information will vary depending on this. For instance, if "show" is used with substitutes like "exhibit", "demonstrate" etc., **c** will move close to words related to the verb reading; by contrast, if the substitutes are "series", "program" etc., **c** will be close to words related to entertainment, television, etc.

3.3.2 Auxiliary Tasks

We define the following auxiliary probing tasks: a diagnostic model – a machine learning model trained merely for the purpose of analyzing another one – is trained to retrieve a word representation out of a LM's hidden state, by learning to match ground-truth representations. To obtain the inputs and outputs of our task, we deploy a LM on text passages from CoInCo, and for each target word annotated with substitutes we extract its processing and predictive hidden states, as well as the representations **l** and **c**. Given these two representations, we define two objective tasks, namely:

- **lex**: to retrieve lexical information; that is, the representation **l**;
- **ctxt**: to retrieve the contextual word information; that is, the representation **c**;

Previous works (Adi et al., 2017; Conneau et al., 2018; Voita et al., 2019) looked at the extent hidden states trace a word input, by looking at whether the identity of the word is retrievable. The **lex** task is related to these approaches but differs from them in that we ask the diagnostic model to retrieve the high-dimensional representation of a word rather than its index in the vocabulary. In the case of processing states, we essentially ask the diagnostic model to reconstruct the input representation starting from which the

[6] With $\alpha = 2$, Recall-10: 0.94; GAP: 0.91.

hidden state was computed, akin to the role of a decoder in an auto-encoder. While the target for **lex** will be the same for every occurrence of the word – the word embedding, it will instead change for **ctxt**, which, as described in the previous subsection, will depend on the contextual paraphrases of the word. Thus, to solve the **ctxt** task the hidden states need some degree of context-sensitivity.

We train diagnostic models D to carry out **lex** or **ctxt**, respectively. Each model is a non-linear transformation, optimized to maximize its promixity – cosine – between its output and a target representation **r**:

$$D(\mathbf{h}) = \text{tanh}\,(\mathbf{h}\,W + \mathbf{b}) \tag{3.7}$$

$$\text{loss} := 1 - \text{cosine}(D(\mathbf{h}), \mathbf{r}) \tag{3.8}$$

What the target representation **r** is depends on the objective task: To carry out **lex** $D_{\mathbf{l}}$ uses the lexical representation **l**. To carry out **ctxt**, $D_{\mathbf{c}}$ uses the contextual representation **c**. We abide by the common practice of keeping the diagnostic model simple and small in parameter size. This is under the assumption that limiting the complexity of the diagnostic model constrains that of the task that the model can learn. It cannot solve the task from scratch or memorize regularities in the data, but only identify the relevant information that is already encoded in the input representation. To give an example with the **ctxt** task, the diagnostic model extracts information relevant to the interpretation of the word – **c** – from a hidden state as the LM got to encode it, and not by computing the interpretation itself on the basis of information in the hidden state. As I discuss in Section 3.5, this assumption can however be questioned.

We train distinct diagnostic models for each objective task (**lex** and **ctxt**) and class of input representations (for each hidden layer; processing and predictive, separately).[7] The CoInCo corpus, with LexSub annotations, comes with a split for development and testing. We further split the development set into training and validation (*train*/*dev*: 80/20%). This leads to 7.9K/2K/5.2K word occurrences as *train/dev/test* data.[8]

3.3.3 Evaluation

For each datapoint, we compute the cosine similarity of $D(\mathbf{h})$ to the context-invariant and contextual word representations, respectively, depending on the task:

$$\text{cos-}\mathbf{l} := \text{cosine}(D_{\mathbf{l}}(\mathbf{h}), \mathbf{l}) \tag{3.9}$$

$$\text{cos-}\mathbf{c} := \text{cosine}(D_{\mathbf{c}}(\mathbf{h}), \mathbf{c}) \tag{3.10}$$

[7] For each diagnostic model, we optimize its hyperparameters based on validation loss. Details in Appendix A.2.

[8] For each LM, only datapoints whose target word and at least one substitute are in the model's vocabulary are covered; for BERT, words that are split into subwords are considered not covered. This yields a coverage of $\approx$ 90% for BERT, and $\approx$ 95% for the biLSTM.

This focuses on the absolute similarity between the extracted representation and the gold one, and corresponds to the metric maximized during training.

For the evaluation on the test data, we also deploy each representation extracted from a hidden state in a substitute retrieval and ranking task, aimed at testing its similarity relations in the space of words. This evaluates the extent to which the representation accounts for the contextual interpretation of the word, as reflected by the set of substitutes S provided by annotators. We follow standard practices for evaluation on the LexSub task. For retrieval, we consider the top 10 lemmas in the cosine nearest-neighbors of $D(\mathbf{h})^i$ as proposed substitutes $\hat{S}$,[9] and calculate recall out-of-ten (McCarthy and Navigli, 2007):

$$\text{Recall-10} := \frac{\sum_{i \in \hat{S}} n_i}{\sum_{j \in S} n_j} \tag{3.11}$$

weighting each retrieved substitute i by the number of annotators that produced it – n_i. In the ranking task, all substitutes of a given word across the full LexSub dataset are pooled and considered as the word-specific set of candidate substitutes. The gold ranking for a word occurrence will be given by the descending order of frequency in the annotation (for each candidate i, n_i). A predicted ranking is obtained ordering candidate substituted by descending order of cosine similarity to the evaluated representation. The resulting ranking $\hat{R}$ is evaluated against the gold one R through generalized average precision (Thater et al., 2009):

$$\text{GAP} := \frac{\sum_{i=0}^{|\hat{R}|} I(i)\ p_i}{\sum_{j=0}^{|R|} I(j) g_i} \tag{3.12}$$

$I(i)$ is 1 if $n_i > 0$ else 0, p_i and g_i are the positions of word i in the predicted and gold ranking, respectively – $\hat{R}$ and R. Contrarily to recall, this metric does not require that the representation is among all words the closest to the substitutes, but just that it is closer to these than non-substitutes candidates, rewarding proximity to substitutes produced by more annotators. Both Recall-10 and GAP are sensitive to the relative order in similarity relations to words but, contrarily to cosine scores, not to their absolute magnitude.

We compare the results to a non-contextual baseline, namely the lexical representation **l** itself (the word embedding). It has trivially a cos-**l** of 1, but when evaluated in cos-**c** or with LexSub scores, it measures how much the contextual interpretation of a word can already be modeled by its embedding alone, without appeal to a contextualization. The content of a word's embedding tends to reflect its distribution over senses (Arora et al., 2018): the most frequent senses are likely strongly represented in the word embedding,

[9] We consider the full vocabulary in the ranking, with exception of non-word characters (excluding subwords, punctuation, numbers and special tokens). We use the Spacy lemmatizer to identify the top lemmas.

thus already accounting for many occurrences of the word (Gale et al., 1992). We can thus expect this baseline to be rather competitive in practice. For LexSub evaluation with BERT, we also report the results obtained using the LM probability scores relative to the target word l (Eq. 3.3). In this case, we rank words not by similarity relations but by their probability score. $P(l)_{\text{proc}}$ is the distribution when the word is not masked, vice-versa for $P(l)_{\text{pred}}$. This baseline offers a non-supervised evaluation of the word information the LM has once gone through all the layers (i.e., at the output), which we can compare to that extracted with diagnostic models.

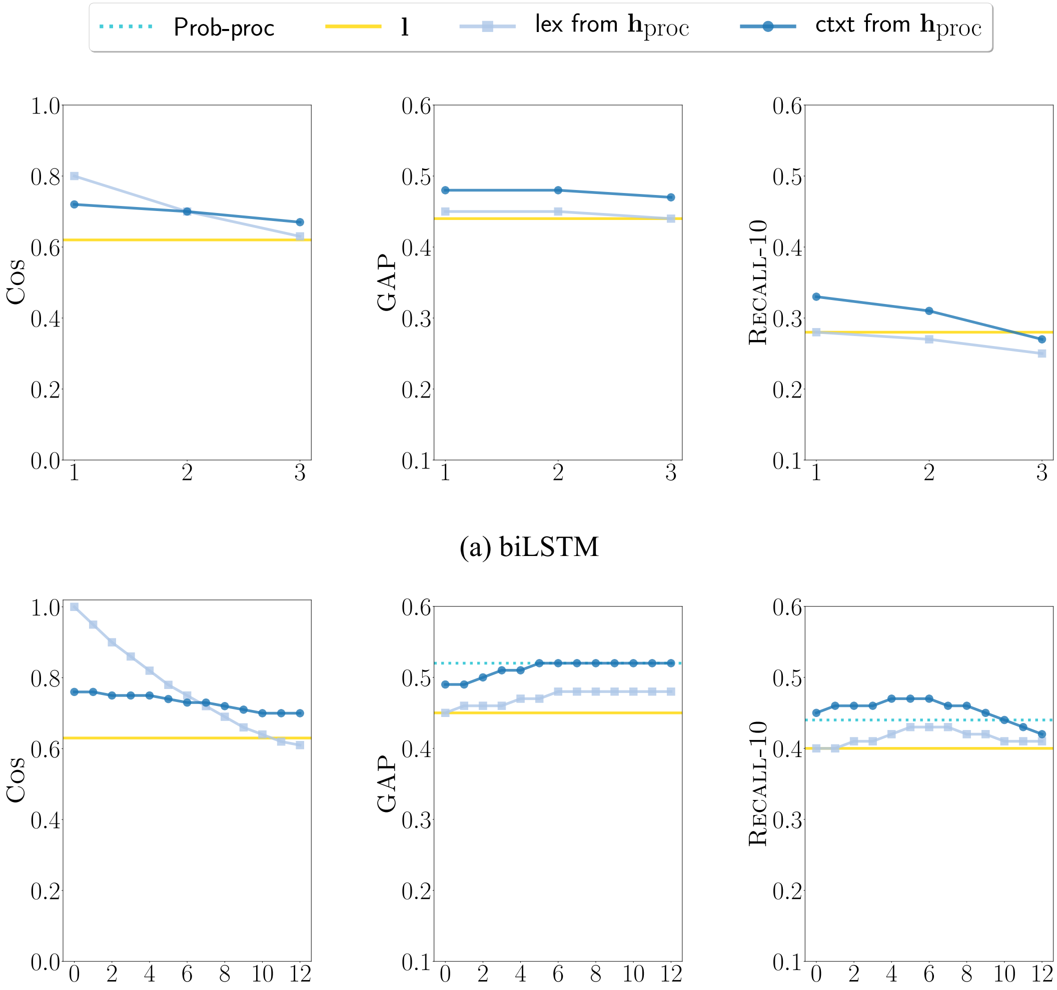
3.4 Results

3.4.1 Word Information in Processing States

In Figure 3.2, I report the evaluation results on the test data of CoInCo for processing states from the biLSTM and BERT, respectively. We separately consider the representations extracted using diagnostic models trained on the **lex** and **ctxt** tasks, and report the average across the dataset for Cos (to **l** or **c** depending on the objective task), GAP and Recall-10. In Table 3.1 I report some examples of substitutes (closest 10 lemmas) for BERT representations.

Cosine. The cosine score was the metric maximized during training, and reflect the extent to which the diagnostic model could extract a representation matching the target one (**l** or **c** depending on the task – **lex** and **ctxt** respectively). From both LMs and tasks, word representations are retrievable with precision – high cosine (in all cases $D(\mathbf{h}) >$ .6; in the case of the **ctxt** task, above the word embedding baseline). This speaks to 1) in the **lex** task, the retention of lexical information, and 2) in the **ctxt** task, the sensitivity to the correct interpretation of a word (i.e., ambiguity resolution). These results suggest that contextualizing the input information does not necessarily entail that only the relevant information is carried over in the hidden states: both lexical and contextual word information seem to be present.

For both LMs and tasks, cosine scores decrease moving towards the output (from the first to the last layers). They do so much more drastically in the case of lexical information. The seeming loss of lexical information could be explained by a progressive modulation of its information: the lexical information is "reduced" to highlight only the contextually relevant information. Indeed in BERT, from around layer 5, we find better retrieval of the contextual representation **c** than the lexical one **l** ($\text{Cos} - \mathbf{c} > \text{Cos} - \mathbf{l}$). However, the fact that cosine scores also decrease moving towards the output in the **ctxt** task ($\text{Cos} - \mathbf{c}$), albeit more smoothly, suggest that contextualization may not alone explain why the cosine goes down across layers.



(a) biLSTM

(b) BERT

Figure 3.2: Cos, GAP and Recall-10 scores on test data for representations extracted with the **lex** and **ctxt** tasks ($D_{\mathbf{l}}$ and $D_{\mathbf{c}}$), respectively, from the biLSTM and BERT processing states. Cos is measured to **l** or **c** depending on the objective task; in the case of the word embedding baseline (**l**), we compute the cosine to **c**.

We consider two explanations, whose impact the next analyses should allow us to disentangle. First, we posit that the negative tendency of cosine towards the output might be due to the objective task, analogously to what was observed by Voita et al. (2019). For instance, causal language modeling (predict the next word) leads to gradually lose focus on the current word, while masked language modeling (predict the word at the current position) leads to stronger retention of input information. In the biLSTM, the latest layers are used to predict a word other than that at the input: it may be beneficial to lose the focus on the current word, as the LM encodes other information that is relevant to the prediction. The same, however, does not apply to BERT, where words are predicted at the output in correspondence to their position at the input (Figure 3.1b). Still, it is possible that, through the layers, as the states account for other information (e.g., neighboring words, syntax, coreference; Tenney et al. 2019a; Klafka and Ettinger 2020), the information about the input word is reduced or at least compressed, making it harder to reconstruct for the diagnostic model. I refer to this hypothesis as H1: *we extract less accurate word information towards the output because this decays across the layers.*

Second, later layers are the result of a more complex transformation of the input (they pass through more layers of computation). Learning to extract a representation that is in the subspace of the input is a harder task, considering equal resources (data and parameters of the diagnostic model). For instance, in BERT if we look at the average cosine between processing hidden states (not transformed by the diagnostic model) and **l** or **c**, these decrease dramatically across layers, thus increasing the cosine margin that the diagnostic model needs to recuperate during training.[10] I refer to this hypothesis as H2: *we find less accurate word information because it is intrinsically harder to extract.*

The LexSub evaluation can help us not only to confirm results regarding the presence of lexical and contextual information, but also potential effects of the hypotheses which we put forward above. This is because they assess how well representations account for interpretations of words independently of the absolute magnitude of cosine scores, which may be decreasing across layers because of the effect of H2 and not only due to decay of word information (H1).

LexSub. I focus first on the general results (and not the patterns across layers). When comparing representations extracted through the **lex** and **ctxt** tasks, respectively, better results are found in the second task for both LMs. In the biLSTM, the extracted representations achieve on average comparable or worse results at LexSub than the word embedding baseline. In BERT, some layers improve a bit over the baseline, but are still worse than in the **ctxt** task. This is in line with the **lex** task objective: to retrieve lexical – possibly underspecified – information, as opposed to the contextually relevant one. The fact that the diagnostic model trained on this task tends to extract a representation that

[10] For BERT, the cosine to **l** goes from .92 to .19 (from layer 0 to 12), to **c** from .53 to .14. The complete results are reported in Appendix A.3

is less context-sensitive confirms the retention of much lexical information in the hidden states. But when using the diagnostic model trained on the **ctxt** task, the retrieved representation reflects contextual substitutes to a larger extent. Results are generally higher for BERT which appears to outdo the biLSTM at the contextualization of word information. Overall, these results confirm that the hidden states tend to be sensitive to the contextual interpretation of words, and that this does not interfere with some degree of retention of underspecified lexical information. Examples of ambiguity resolution in BERT can be found in Table 3.1, showing predicted substitutes (10 closest lemmas) for representations extracted from the hidden states.

The biLSTM and BERT results diverge when looking at the LexSub scores across layers. For the biLSTM, both Recall-10 and GAP decrease towards the output, just like Cos did. The best level of contextualization is found at the first layer, suggesting that this LM focuses on lexical ambiguity resolution first to then presumably move on to attend to other information that is relevant to the output prediction. This seems to support H1 for the biLSTM: in processing states, there is less word information moving towards the output. The same is not found in BERT, where the picture is more complex and suggests that the quality of word information does not actually degrade moving towards the output, in spite of the cosine patterns. In terms of GAP, results progressively improve across layers, eventually resulting in a plateau. Recall-10 initially increases, reaches its peak at layers 4-6, to then decrease towards the output (though staying always above the word embedding baseline). This suggests a contextualization mechanism that initially benefits the representation of the correct interpretation, but then results in worse behavior in the latest layers when focusing on retrieval of substitutes. We investigate this further in the next section, considering a potential effect of masked language modeling.

$P(l)_{\text{proc}}$ – the probability distribution over words computed at the output – can be seen as the final behavior of the LM after all the processing in the hidden layers. We find that in the LexSub tasks using $P(l)_{\text{proc}}$ performs comparably to the last few layers in GAP and a bit better on Recall-10 (though still worse than layers 4-6). If taking the latter metric as reference, this supports the aforementioned idea that a bit less precise ambiguity resolution is displayed in the layers close to the output. But also, since the representation extracted from the last layer – from which the probabilities are computed (Eq. 3.2-3.3) – is a bit worse, it suggests that either the diagnostic model does not manage to fully extract all the word information there is in the state (due to limited capacity or training data), or that the final steps in the network to compute the probabilities aid the retrieval of correct substitutes.

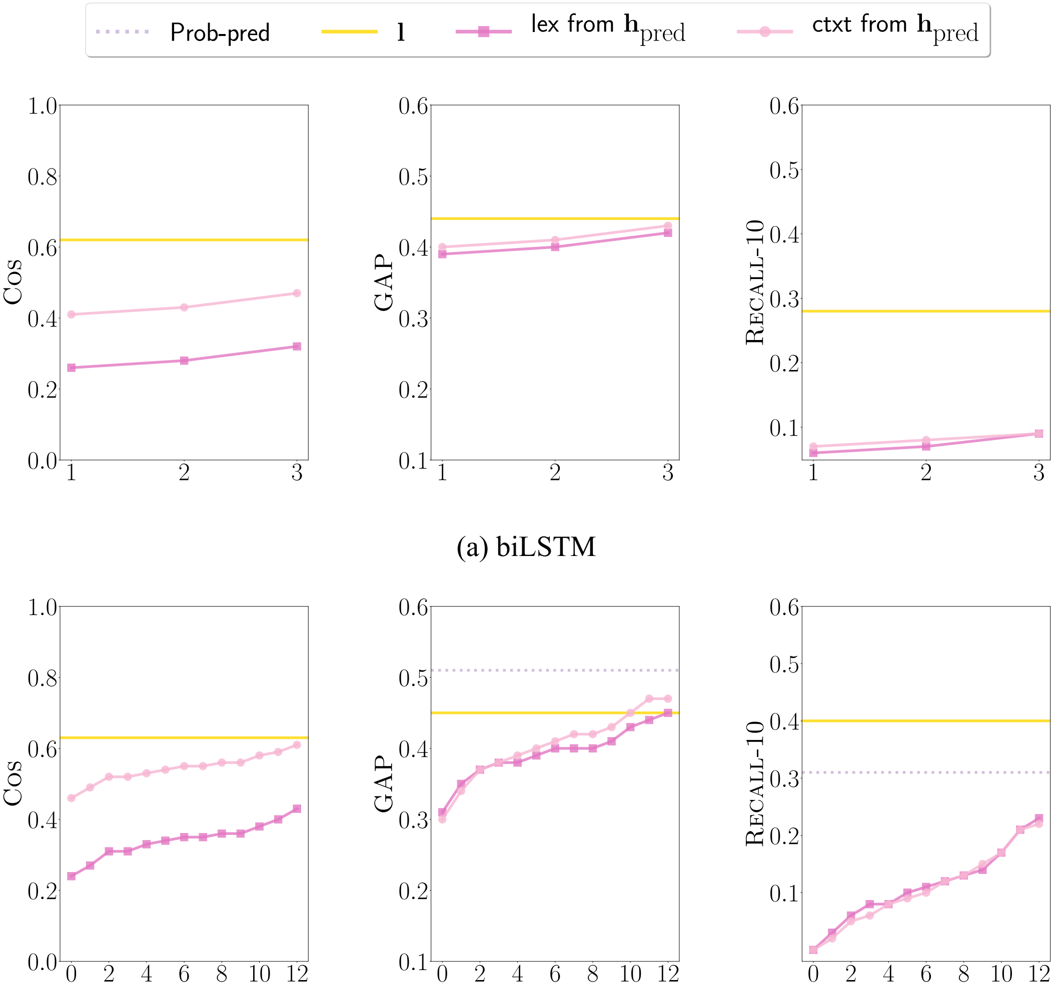
The fact that in BERT cosine patterns do not match those of LexSub scores suggests that looking only at cosine – the metric maximized in training – for evaluation can lead to an unfair comparison across hidden states. The fact that across layers lower cosine scores are found does not mean that hidden states progressively get to encode less, or less correct, word information. In fact, intermediate layers (4-6) are found to account for word information – considering both Recall-10 and GAP – better than the other layers. This is because a representation may be further in absolute terms to the embedding of its substitutes, but yet exceed at ranking them correctly in terms of the relative ordering of cosine similarities. The cosine magnitude seems to be speaking more of the difficulty of extracting a representation close enough to the right space (H2) than to the amount and quality of word information in a hidden state (H1). This underscores the importance of computing LexSub scores as part of the analyses, since, as in the case of BERT, they may not necessarily align with the cosine evaluation.

3.4.2 Word Information in Predictive States

Contrarily to processing states, predictive states are computed without the LM observing the target word as input but when the word is to be predicted at the output. What we are evaluating is therefore the information about the target word that the LM gets to represent, presumably because this is identified to be useful for the word prediction. Therefore, we expect: 1) in comparison to processing hidden states overall less word information (because of uncertainty over the word itself), 2) more word information moving towards the output (approaching the prediction), and 3) more contextual word information than lexical one (due to being predicted solely based on the context).

This scenario is confirmed by our results on the test data of CoInCo, reported in Figure 3.3 for predictive states from the biLSTM and BERT. The cosine patterns are opposite to those found for processing states and identical between the two LMs: the target representation – **l** or **c** – can be predicted with more precision moving towards the output. This makes sense as it is at the output that the LM computes a word prediction, leading to accumulating more and better information about the word to predict across layers. Performances in the **ctxt** task are higher than those on the **lex** one, though worse in both cases than those of processing states. This is in line with expectations as predictive states are context-sensitive by design (they can only predict information based on the context), and never access the underspecified information in a word embedding. Therefore, while they may be uncertain about the word to predict, they can only predict aspects of this that are contextually relevant. For instance, in example (3) in Table 3.1, where "glass" is used as 🥛, the predicted information has to do with containers and not the sense of glass as material, which is not expected given the context.

The LexSub scores essentially confirm these results. Again, the results improve towards the output, confirming that word information becomes more reliable across the layers, likely due to decreasing uncertainty about the word to predict. Finally, representations extracted in the **lex** and **ctxt** tasks are comparably good at retrieving and ranking substitutes. This can be led back to predictive states containing only word information that is predicted to be contextually relevant: even when probed to extract information relevant to multiple senses (**lex**), the diagnostic model can only retrieve context-sensitive information.



(a) biLSTM

(b) BERT

Figure 3.3: Cos, GAP and Recall-10 scores on test data for representations extracted with the **lex** and **ctxt** tasks ($D_{\mathbf{l}}$ and $D_{\mathbf{c}}$), respectively, from the biLSTM and BERT predictive states. Cos is measured to **l** or **c** depending on the objective task; in the case of the word embedding baseline (**l**), we compute the cosine to **c**.

	(1) ... You could write a **play** about someone like him. substitutes: *script, show, comedy, drama, line, manuscript, performance ...*
l	*work, game, dance, perform, player*
$\mathbf{h}^{6}_{\text{proc}}$	*show, book, story, program, novel*
$\mathbf{h}^{12}_{\text{proc}}$	*show, project, document, act, program*
$\mathbf{h}^{12}_{\text{pred}}$	*story, presentation, treatise, tragedy, movie*
	(2) ... Sales for the **full** year were \$6.6 billion substitutes: *entire, whole, complete, total, overall*
l	*fully, complete, whole, entire, total*
$\mathbf{h}^{6}_{\text{proc}}$	*large, complete, whole, entire, total*
$\mathbf{h}^{12}_{\text{proc}}$	*first, complete, second, main, major*
$\mathbf{h}^{12}_{\text{pred}}$	*previous, prior, preceding, subsequent, last*
	(3) ... I waited, and drank a **glass** of MorningAfter and three cups of coffee substitutes: *cup, bit, mug, pint, serving, share, shot, swallow*
l	*stone, ceramic, steel, bottle, porcelain*
$\mathbf{h}^{6}_{\text{proc}}$	*drink, beverage, bottle, beer, cup*
$\mathbf{h}^{12}_{\text{proc}}$	*beverage, drink, vodka, sip, bottle*
$\mathbf{h}^{12}_{\text{pred}}$	*chunk, container, drink, bottle, snack*
	(4) ... and suddenly the forgotten, in revenge, rears up to **savage** the unwary. substitutes: *terrorize, attack, blast, crucify, destroy, disrupt, distress*
l	*vicious, brutal, ruthless, fierce, monstrous*
$\mathbf{h}^{6}_{\text{proc}}$	*vicious, hideous, horrific, monstrous, seduce*
$\mathbf{h}^{12}_{\text{proc}}$	*destroy, seize, eliminate, obtain, weaken*
$\mathbf{h}^{12}_{\text{pred}}$	*destroy, assist, protect, oppose, confront*

Table 3.1: Examples of predicted substitutes (5 closest lemmas) for target word occurrence (in bold) of representations extracted from BERT layers using the diagnostic model from the *ctxt* task; **l**: word embedding.

3.4.3 Processing as Result of Prediction

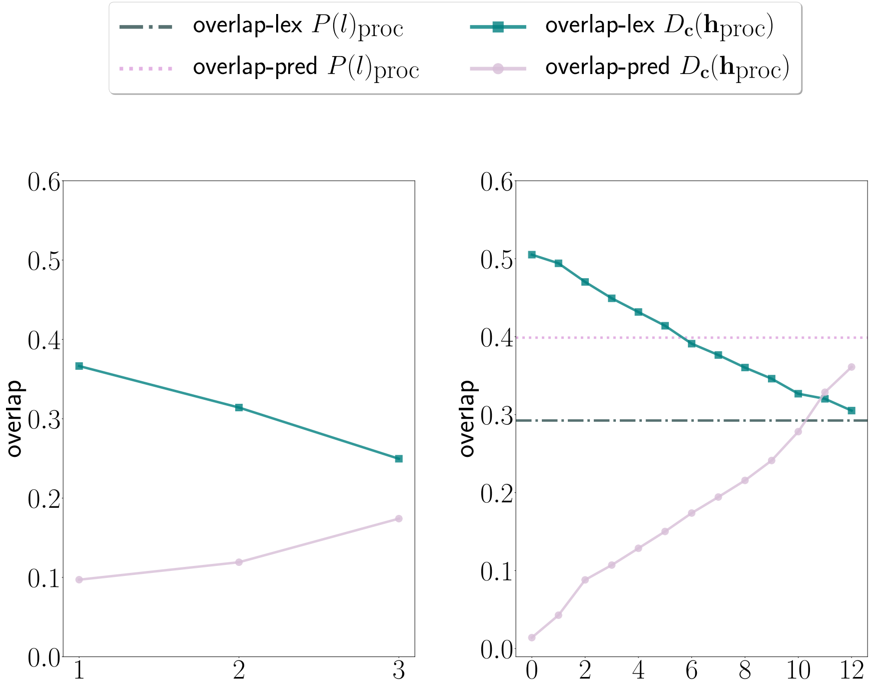
In Section 4.4.1, we observed that in BERT the contextual interpretation of a word tend to improve across the layers, but – when looking at Recall-10 and thus the proximity to substitutes – decaying a bit towards the output (Figure 3.2). This suggests that during processing the states are subject to a mechanism that tends to initially benefit the representation of contextual word information (better in the central layers) but then, applied across the layers, eventually results in suboptimal behavior. We hypothesize that this is an effect of the objective task – masked language modeling – intertwining mechanisms for processing and prediction of words.

In BERT, a word prediction is computed at the output in correspondence with its position at the input (Section 4.2), both when the word is masked or given as input. The difference in the input is therefore the sole aspect that distinguishes predictive and processing states (Figure 3.1b). We can thus think of processing and prediction of word information in BERT as stemming from a unique mechanism. The same cannot be said for the biLSTM where processing and predictive states come from different positions in the sequence and are used to predict different output words (Figure 3.1a).

We posit that word information in BERT's processing states results from a friction between processing the current word, and predicting it as if the word had not been seen. Lexical ambiguity resolution in processing states would then occur as a result of, on the one hand, attending to the inputted lexical information, and, on the other, inferring information that would be contextually relevant as if the input word had been masked. The latter is the information that would be represented if this was a predictive state, instead of a processing one. We can conceive this as contextual expectations formed by the LM about the content relevant for that slot in the input. As contextual expectations improve towards the output (as shown by the results of predictive hidden states), it is plausible that BERT would initially mostly focus on the inputted lexical information to then increase the weight put on the contextual expectations as they become more reliable across layers. Contextualization in a layer would then depend on a tug of war between lexical information and contextual expectations.

Concretely, this hypothesis predicts that, as we progress towards the output, the contextual information in processing hidden states shares less with the inputted lexical information, and more with the counterpart predictive states. Under this projection, the layers closer to the output would be the ones that are the most affected by contextual expectations. This hypothesis could clarify how BERT resolves ambiguities, but also explain why we found contextualization, in terms of Recall-10, to initially improve across layers to then worsen close to the output: Though expectations constitute a good context-sensitive bias, attending to them above a certain degree may ultimately be misleading, as they are based on uncertainty over the target word.

To test this hypothesis, for each processing state at layer i, we look at the representation extracted in the **ctxt** task – $D_{\mathbf{c}}(\mathbf{h}_{\text{proc}})^i$ – and its candidate substitutes $\hat{S}^i_{\text{proc}}$ (i.e., the



(a) biLSTM

(b) BERT

Figure 3.4: Overlap of predicted substitutes (10 closest lemmas) of representations extracted from processing states (with **ctxt** task) with lexical information and the predictive state counterparts, respectively.

10 closest lemmas to the representation). We compute the overlap of these substitutes from the ones of the input lexical representation $\hat{S}_{\text{lex}}$, on the one hand, and the predictive counterparts $\hat{S}^i_{\text{pred}}$ (from $D_{\mathbf{c}}(\mathbf{h}_{\text{pred}})^i$), on the other. To avoid confounders, we ignore the intersection of $\hat{S}^i_{\text{pred}}$ and $\hat{S}_{\text{lex}}$, focusing on substitutes that are uniquely shared with one of the two sets.

$$\text{overlap-lex} := \frac{\hat{S}^i_{\text{proc}} \cap \ (\hat{S}_{\text{lex}} - \hat{S}^i_{\text{pred}})}{|\hat{S}^i_{\text{proc}}|}; \tag{3.13}$$

$$\text{overlap-pred} := \frac{\hat{S}^i_{\text{proc}} \cap \ (\hat{S}^i_{\text{pred}} - \hat{S}_{\text{lex}})}{|\hat{S}^i_{\text{proc}}|} \tag{3.14}$$

The results per layer are displayed in Figure 3.4. For comparison, we also report the results obtained for the biLSTM. In line with the projections from the aforementioned hypothesis, for BERT moving towards the output word neighbors from processing states overlap less with those from the input word embedding, and more with those of the counterpart predictive states. For the biLSTM, we find the same but the overlap with substitutes from predictive states is much smaller. It may result simply from predictive and processing states concurrently focusing on analogous information, since we know the two are derived from disjoint mechanisms. We instead take the results from BERT as pointing to a progressively increasing reliance on expectations across the layers at the expense of lexical information, justified by the fact that prediction and processing of words in BERT are not independent mechanisms.

These results, together with those displayed in Figure 3.2b, suggest that the optimal degree of reliance on the context is achieved in central layers: attending to contextual expectations can positively guide the interpretation of a word, but attending to them above a threshold seems not to be optimal. It is important to notice, however, that this is a conclusion based on focusing on average Recall-10 results across layers: First, measuring lexical ambiguity resolution with other metrics need not provide the same picture. For instance, GAP results (3.2b) show different trends. Second, while a certain degree of reliance on expectations may be beneficial to account for substitutes across the dataset, based on a qualitative analysis, different occurrences of words may be better modeled focusing more or less on expectations. For instance, examples (2) and (4) in Table 3.1 are cases where focusing more on expectations leads to accounting better for the interpretation of the word (comparing layer 6 to 12).

3.5 Summary and Discussion

In this chapter, I presented a method to investigate the word information encoded in a LM's internal activations, and applied it to study the inner dynamics of two LMs. This allowed us to inspect both the abilities and strategies of LMs in resolving lexical ambiguities. We compared hidden states obtained observing the target word as input – processing – to those that are agnostic of the word's identity as they have to predict it at the output – predictive. I discuss here the findings, focusing first on their implications for our understanding of LMs, and then reporting some methodological considerations.

3.5.1 Lexical Ambiguity Resolution in Language Models

Representation of Contextual Information

Our experiments investigated the degree to which LMs infer and encode the correct interpretation of a word. Our results are in line with other studies (Reif et al. (2019); Nair et al. (2020); Garí Soler and Apidianaki (2021), a.o.) reporting that LMs have sufficiently context-sensitive internal representations to well account for situation-specific interpretations of words. We here attempted to shed light on the ambiguity resolution strategy leading to this behavior in terms of representation of word information in the hidden states. In particular, we studied how the representation of contextual information varies throughout the depth of the network (i.e., across layers), and how processing states – where ambiguity resolution should, if anywhere, occur – compare to predictive states.

While the biLSTM seems to select an interpretation to then presumably focus on other information relevant to the output prediction (better results at layer 1), BERT, a deeper model, seems to apply a more progressive ambiguity resolution across the layers (best results from layer 4). In accounting for contextual interpretations of words, BERT's hidden states outdo the biLSTM's ones, though in general there is some margin for improvement. The two LMs differ in size and amount of training data (BERT surpasses the biLSTM in both), a plausible explanation of the differences in results. We however also identified an aspect in the BERT architecture possibly impacting the way it resolves lexical ambiguities. In BERT, processing and prediction of a word are intertwined, with the two merely differing by the token passed as input (whether it is the word itself or the `[MASK]` token). This seems to push the LM, when processing an input word, to attend not only to its lexical information but also to the information that would have otherwise been predicted for that slot, had the word not been seen as input. We thus posit that masked language modeling creates a bias for an expectation-driven resolution of ambiguities. That incorporating such a bias can be beneficial for ambiguity resolution is further supported by linguistic accounts highlighting the role of expectations from both the pragmatic and processing perspective (Wilson and Sperber, 2006; Clark, 2013).

We found evidence that different hidden layers in BERT pay different degrees of attention to lexical information and expectations from the context, respectively. This affects the extent to which a certain layer accounts for the interpretation of the word, at least as evaluated in terms of substitute retrieval. On average, we found central layers, taking into account both the lexicon and the expectations to an intermediate degree, to lead to the best results. Yet, qualitatively looking at different word occurrences, we can note that in certain situations other states, incorporating more or less towards expectations, are more apt to explain a word's interpretation. These results highlight the need to more directly inspect the interplay between lexical information and contextual expectations as an explanatory mechanism in word interpretation. We address this in Chapter 4 where we define a computational framework for the study of word interpretation, leveraging representations from LMs. In particular, we build on the results from Section 3.4.2 as support for deriving proxies for contextual expectations about word content using LMs.

Retention of Lexical Information in Processing

Our results indicated that for both LMs, lexical information – potentially associated with multiple senses – tends to be recoverable from processing states to a large extent. This, however, does not seem to obstruct contextualization: we found retention of lexical information even at layers where contextual word information could also be retrieved well.

From an NLP perspective, using internal activations from LMs for transfer learning proved to be a successful strategy in previous work, leading to vast improvements across many tasks (Peters et al., 2018a; Devlin et al., 2019). It is common to link the gains of this approach to the use of context-sensitive representations of words – hidden states from LMs – potentially solving the issue that word embeddings encode information relevant to multiple senses (the *meaning conflation deficiency*, as referred to in Camacho-Collados and Pilehvar 2018). Our results indicate that, though these representations tend to reflect contextual interpretations of words to a large extent, they do not fully get rid of the underspecification in the input word embedding, as much lexical information is retrievable. However, this does not need to cause issues when the representations are used for transfer learning, as long as contextual information is also encoded, enabling context-sensitive behavior. Besides, the retention of lexical information may even be useful in case the LM did not correctly resolve the ambiguity, so that the new model receiving the hidden state as input can get a second chance at it.

We can compare the ambiguity resolution mechanism of the LM, emerging directly from data, in the way it relates to what is known about human processing of lexical ambiguities. Several accounts of word meaning access argue for initial activation of information relevant to multiple senses of a word (a.o., Duffy et al. 1988; Frisson and Pickering 1999; Vitello and Rodd 2015). After this initial simultaneous activation of diverse information, humans do not seem to maintain irrelevant sense alternatives for long and move on to rapidly select an interpretation (Tanenhaus et al., 1979; Duffy et al., 1988; Rodd et al., 2010), suppressing the information that was initially activated but turned out to be ultimately irrelevant. At a general level, also in LMs underspecified lexical information, potentially relevant to multiple senses, is initially activated when passing a word embedding as input. Our results suggest that this remains largely present (activated) in the hidden layers until the output, independently of whether an interpretation gets selected. This suggests that the initially activated but non-selected information is not entirely suppressed in the activations of the LM after ambiguity resolution, while in humans it tends to decay rapidly after. This seeming difference between humans and LMs should however be taken with a grain of salt: from the bidirectional LMs we analyzed we cannot infer a time-course of ambiguity resolution which we can then compare with that posited for humans.[11]

A general question, encompassing both the linguistic and the more applied NLP perspective, is why a LM would acquire such a mechanism to ambiguity resolution:

[11] It is not clear what "rapid", said of selection or suppression of information, means for a bidirectional deep learning model, given 1) that both sides of the context are considered here to process a word, and 2) that we cannot map the depth of processing in a LM (the hidden layer) to a specific temporal stage (e.g., is rapid selection the inference of an interpretation at layer 2 or 6?).

that is, to carry over much, possibly noisy, lexical information, independently of the inference of an interpretation. Since this is a mechanism learned from data, it must have been identified as part of training as somewhat useful for the objective task. I below put forward some motivating hypotheses, which, though not testable with the current results and methods, I hope will spur further research on this topic.

Safety Mechanism. During training the LM recognizes, on the one hand, the need to disambiguate word information– at the expense of possibly being incorrect – and, on the other, that the inputted lexical information is often, as is, a safe choice. First, frequent senses will already have a large representation in the word embedding; second, different senses of a word are often related (e.g., "glass" as a material or a container of that material), thus sharing a common core. As a response, the LM would find it beneficial across cases to keep track of the lexical information "just in case", for instance, to smooth the effect of misinterpretations (Blott et al., 2020). The model can then recover more easily, or at least limit the damage, if information relevant to other interpretations is, to some extent, still activated in the states.

"Good enough" Representations. A motivation for our experiments was that LMs are expected to develop a strategy to cope with lexical ambiguities. However, often a superficial resolution of ambiguities, or even lack thereof, may be sufficient for the objective task (word prediction), without requiring a more detailed analysis of the input. This may be due to the monosemy of some words and distinctions among related senses (as the example of "glass" above). There may be then somewhat contradictory incentives to, on the one hand, resolve ambiguities and, on the other, "lazily" just pass forward the underspecified inputted information as is. As a result, the model would learn to retain much information from the input, despite the noise it may contain. This hypothesis is inspired by "good enough" accounts of linguistic processing (Ferreira and Patson, 2007; Frisson, 2009), building on evidence that humans do not always derive complete and accurate interpretations of an input.

These two hypotheses that I have described justify the retention of lexical word information, assuming that this information is encoded in hidden states in parallel to contextual word information. The following hypothesis considers instead a different scenario.

Implicit Ambiguity Resolution. Under this hypothesis, the LM does not actually explicitly resolve lexical ambiguities in the hidden layers: the content of the inputted word embedding is not modulated to the context and the hidden states do not get to consequently encode contextual word information (but only lexical information). At a first glance, this idea may seem at odds with our results, as well as previous research providing evidence for the context-sensitivity of LMs to word senses. This explanation is however not contradictory with the existence of such context-sensitivity, but simply puts forward a different way in which this would be achieved: in an "implicit" way.

Instead of contextual word information, the hidden states would simply represent: 1) the lexical information about the word, as retained from the input layer, and 2) information from its co-text. Because different senses of a word tend to appear with different context cues, the hidden states relative to occurrences of different senses will end up being different from each other, despite sharing word-specific lexical information. To give an example, the representation of "mouse" when used as 🖱️ will be different from that of an occurrence as 🐭, because their contexts will be, in the first place, different. The variability of representations of the same word type, depending on the context, may just be sufficient to enable a context-sensitive behavior, both in the predictions of the LM at the output (i.e., words relevant to the correct interpretation) and when using these internal representations – for analysis or transfer learning – to predict or represent word senses (Loureiro et al., 2021; Garí Soler and Apidianaki, 2021). Models can associate different output behaviors with different representations, and differences in representations can correlate with graded differences in intended meaning. This hypothesis is, in comparison to the others previously described, more negative regarding the ambiguity resolution abilities of LMs: They do not have to develop an ad hoc mechanism for ambiguity resolution, because representing word occurrences through their linguistic context is on its own sufficient to reflect differences in word senses.

Under this hypothesis, our results are explained if: 1) lexical word information is encoded in hidden states, and therefore it can be extracted by the diagnostic models; 2) contextual word information is not encoded, but it can still be retrieved. This is possible if, leveraging the differences of hidden states across word usages, the diagnostic models learn to *compute*, as opposed to *extract*, representations of word interpretations; for instance, by modulating the lexical information encoded in the hidden state based on the context. Our diagnostic models were purposely kept simple and small so that they did not have the capacity of learning such a complex mechanism. Yet, as some supervision is after all involved, we cannot rule out that this explanation may have at least a partial role on the results. This suggests some methodological caveats, which I discuss in the next section.

3.5.2 Methodological Considerations

This chapter adds to a long list of recent work analyzing the dynamics of deep learning models and in particular LMs (Belinkov and Glass, 2019). These can be roughly clustered in approaches focusing on the information encoded in the internal activations of a model (e.g., whether the representation encodes the identity of a processed word; Adi et al. 2017), and those which instead test the model's behavior at the level of its output predictions (e.g., whether the LM predicts the verb with correct number agreement with the subject; Linzen et al. 2016). Our experiments fall in the first category as we inspected the word information contained in hidden states defining auxiliary supervised probing tasks. Other studies reporting contextualization of word information in LMs also focused on information in the hidden layers, by, for instance, carrying out a classification task with these representations as input (Pilehvar and Camacho-Collados, 2019; Reif et al., 2019) or looking at the graded variation of these vectors across contexts of use of a word (Garí Soler et al., 2019b; Ethayarajh, 2019; Nair et al., 2020).

Focusing on analyzing the LMs' internal representations is especially motivated by the fact that these are often used as input representations to other models (for transfer learning): this practice underscores the need for clarifying the linguistic phenomena they account for. However, looking at representations, and specifically relying on auxiliary supervised tasks, also has some challenges, as we identified in our study. How to evaluate the information contained in a high-dimensional vector with uninterpretable dimensions is not straightforward, and requires the definition of rather complex methodologies (supervised or not). This is not an issue per se, but it increases the risk of deriving conclusions that may be to some degree affected by the analysis method itself rather than only by the studied LM.

First, when using supervision we cannot rule out the possibility that the diagnostic model does not simply extract information that is already encoded in the vector, but instead computes it (applied to this study, the aforementioned *Implicit ambiguity resolution* hypothesis), or even simply "refines" it. This may be enabled by leveraging other useful information in the representation and regularities in the data. More frequent word types and, for a certain word, more frequent senses would have an advantage, as there would be more training instances from which to learn. This issue cannot be bypassed easily: Since both the occurrences of words and their senses follow a Zipfian distribution (Zipf, 1949), any corpus intended to reflect the true distributions, like the CoInCo corpus (Kremer et al., 2014), leads to unbalanced sense distributions. There is therefore a hard trade-off between coverage of words and their senses, and reducing potential biases from regularities in the data.

Second, we noticed that some cautiousness is required when comparing results across representations of different types (e.g., from different hidden layers), for which the auxiliary task may be intrinsically harder to start with. Concretely, we cast the supervised probing using a representation learning setup; however, the cosine margin between the target representation and a hidden layer increases across the hidden layers, making the transformation to be learned more complex. As we showed, achieving a lower final cosine between the target and predicted representation does not mean that the task was not learned, and actually in terms of relative similarities a representation with lower cosine could actually be better at accounting for a word's interpretation. Still, this issue may lead to confounders in the comparison. Are intermediate states truly better at contextualization? Or - since mapping them to the word space is easier - the diagnostic model could allocate more resources to, for example, learning more regularities in the data to solve the task?

Finally, even when not using supervised auxiliary tasks, it is hard to actually fully rule out the *Implicit ambiguity resolution* hypothesis when looking at a LM's internal representations. As long as a representation type varies enough across occurrences of different word senses, the analyzed model does not necessarily need to have learned a mechanism to compute the interpretation of a word. Rather it may just have learned a way to represent and compress well the information in the input (Pimentel et al., 2020b). For applications, this need not be a problem, as the representation can still be rich enough in information to allow for successful use. But if our goal is to understand how LMs resolve ambiguities, we need to assess with clarity that they explicitly resolve them – they select an interpretation and comply with this.

A behavioral evaluation of LMs regarding the resolution of ambiguities – including beyond lexical ones – could help in clarifying this aspect. Instead of focusing on the internal representations, one would look instead at the output behavior of the model – its probabilistic output – on data targeting the phenomenon of interest (e.g., Marvin and Linzen 2018; Ettinger 2020). Our experiments included some behavioral analysis when evaluating the probability scores output by the LM in the LexSub task, providing evidence that indeed this reflected a word's correct interpretation to a good extent. Though only looking at the surface behavior of the model and not directly at its inner workings, behavioral analyses of LMs are arguably a more reliable and natural way of assessing the way a LM handles a certain phenomenon, as they involve deploying a LM in the task it was trained to start with (to output word probabilities). In Chapter 5, we make a step towards this direction by defining an evaluation procedure to estimate the uncertainty of a LM over interpretations of an ambiguous portion of a sentence, focusing on syntactic ambiguities (including lexical Noun/Verb ambiguities). I there also discuss the challenges and potentials of applying this method beyond ambiguities at the syntactic level.

Chapter 4

MODELING THE INTERACTION BETWEEN CONTEXTUAL EXPECTATIONS AND THE LEXICON IN WORD INTERPRETATION

The basic implication of lexical ambiguity – that a word may convey different information depending on the context – is that words are to be understood taking into account the context they are embedded in. Consider the following examples:

(19) The architect created an amazing *bridge*. 🌉

(20) The songwriter created an amazing *bridge*. 🎵

While it is not inconceivable that a songwriter designed a structural bridge nor that an architect would write a song section, the context creates strong expectations for each of the occurrences of the word "bridge" in (19-20): an architect is expected to design buildings and such, and a songwriter to create music. Expectations alone however do not determine an interpretation, as not everything that is expected actually could be an interpretation of "bridge". The word itself carries a series of meanings it is most typically associated with, and it would be non-sensical to interpret it as things that are expected in the context but not compatible with the word (e.g., as referring to a tower in (19) or a music album in (20)).

Hence, there are essentially two sources of information which we can use to interpret a word: On the one hand, we can use information about the type of content the word typically expresses, with the various meanings possibly differing in their strength of association to the word. On the other, we can use the context the word is embedded in – linguistic and extra-linguistic – to consider what content is more likely to be intended. Reasoning about these two sources of information, we get to privilege a certain interpretation of the target word occurrence. This basic assumption is, in one form or another, at the basis of most approaches to lexical ambiguity resolution, or subcases of it (e.g., polysemy, homonymy, figurative language, etc.). In this chapter, we test this idea at scale as a general explanatory mechanism for lexical ambiguity resolution, by turning the posited mechanism into a computational framework. Through this framework, we build computational models implementing different degrees of reliance on the lexicon and on contextual expectations, respectively, during word interpretation, and study what interplay between these two sources of information best explains human interpretation judgments.

To carry out these experiments, we use representations from pre-trained neural language models (LMs) as proxies for both the lexicon and contextual expectations. In the previous chapter, I studied the strategy that these models themselves use to resolve word-level ambiguities. In this chapter, I do not focus anymore on analyzing the inner workings of LMs, but rather on the way we can use these models to automatically estimate certain types of information in order to instantiate and test linguistic hypotheses. Concretely, we use representations from LMs to assess the explanatory breadth of lexical disambiguation strategies. Crucially, we do not use the activations reflecting how the LM itself has interpreted a certain word, but rather those representations that can act as proxies for the information we input to our computational model of interpretation: lexical information and contextual expectations.

4.1 Approach and Research Questions

Accounts of word interpretation in linguistics and cognitive science vary across several dimensions (Falkum and Vicente, 2015; Rodd, 2020). To start with, how out-of-context information about a word is stored in the mental lexicon is subject to debate (Klein and Murphy, 2001; Klepousniotou, 2002; Frisson, 2009): whether information relevant to different senses of a word is separately stored, or blended in a unique lexical representation. To derive interpretations, different mechanisms can be posited, such as lexical rules (Asher, 2011), pragmatic inferences (Wilson and Sperber, 2006), or activation of information during processing (Duffy et al., 1988). Despite these differences, we consider that there are common general assumptions:

1. Some context-invariant **lexical information** comes into play when interpreting words, pointing to familiar usage types;

2. The context of a word influences an interpretation by creating **expectations** about what content the interlocutor is more likely to intend to convey, or – else put – that is supported by the context.

We here study how much of lexical ambiguity resolution can be explained if concretely instantiate this assumption into a model where lexical and expected information are combined to compute an interpretation.

Accounts of lexical ambiguity resolution, or subcases of it, remain to date difficult to test at scale. One of the main challenges is that of identifying an appropriate representational format, or meta-language, to represent the inputs to the interpretation process (the lexicon and the context), such that it allows 1) to cover a wide range of words and contexts, and 2) allows for the definition of formal operations that transform a hypothesized mechanism of ambiguity resolution into a precise computational model. The use of computational data-driven methods can help in this challenge if providing reliable proxies for the information to reason about. We in particular turn to LMs to expedite this investigation, by using them to represent information in our computational models.

Specifically, we represent lexical information through the word embeddings in a LM, as an instance of distributional semantics (Lenci, 2008; Boleda, 2020). This framework was often used in previous research as a methodology to approximate the information encoded in a lexicon. As argued earlier in this thesis, distributional representations can be seen as underspecified representations of lexical content, encompassing information relevant to different senses of a word. Their similarity relations reveal information generically associated with a certain word, abstracting over its diversity of usages. For instance, for “mouse”, we can expect the word embedding to be close to both words having to do with 🐁 (e.g., “rat”) and 🖱 (e.g., “keyboard”); for “watch”, both the sense of the verb 👀 and that of the noun ⌚ should be encoded. In principle, to represent the lexicon we could employ alternative methods to represent a distributional semantics perspective (e.g., Pennington et al. 2014; Grave et al. 2018), other than the embeddings from a LM. However, as we will see, using embeddings from a LM simplifies our framework in that, it allows us to have compatible representations of expectations and the lexicon in the same high-dimensional space. In terms of the correspondence of these representations with a certain take on the lexicon, they more closely align to a view where the typical senses of a word are not separately represented, but “compressed” in a multi-faceted representation reflecting the diversity of information associated with that word (Pustejovsky, 1995; Frisson, 2009; Carston, 2012).

We model contextually expected information building on a long tradition in Cognitive Science and Computational Linguistics employed LMs to model word-level expectations (Levy, 2008; Frank et al., 2013; Armeni et al., 2017). Typically, what is used is the probability distribution over the vocabulary output by LMs, to extract, for instance, surprisal estimates. By contrast, we focus on a different way of encoding expectations: a representation that shares the representational format (a vector in the same space) of a word embedding – the lexical representation – and encodes how expected a certain word is in terms of its similarity to the word embeddings. Analogously to how the lexical representation provides a pointer to the information that is typically associated with the word, this representation of expectations can be seen as pointing to the content – potentially diverse – expected by a listener to be conveyed by the speaker given the context of utterance. Such expectations may reflect both syntactic and semantico-pragmatic aspects: For instance, in (21), contextual expectations relative to "watch" should be closer to verbs than to nouns given structural constraints making these more likely (e.g., "take", "open", "keep"). In the context of (21), expectations should not only be close to nouns but also specifically words for objects that often appear on desks (e.g., "laptop", "keyboard", "pen").

(21) Could you please *watch* my bag for me? 👀

(22) I left my *mouse* on the desk. 🖱

In this chapter, I argue and show that under certain conditions a representation of expectations with these properties can be identified in a LM, without the need to introduce any ad hoc training or additional procedure (as opposed to, for instance, how we extracted word information from hidden states in the previous chapter).

We thus have ways of obtaining proxies for both the lexicon and contextual expectations, by simply deploying off-the-shelf a pre-trained LM. The two sources of information have complementary properties: one focusing on information suggested by the word without seeing the context, the other on information suggested by the context without seeing the word. Both rely on a certain degree of uncertainty: both the information that is associated with a word and that is expected in context can be quite diverse. A good representation of the contextual interpretation of a word should locate in a point in the high-dimensional space that overcomes those uncertainties. We define operations that put together the representations of lexical and expected information to reach an interpretation: the content which is in line with both what is typically associated with a word and what is contextually expected. This approach is similar in spirit to vector operations proposed to contextualize distributional semantic representations (Erk and Padó, 2008; Thater et al., 2011). Figure 4.1 sketches out our framework's architecture. For instance, in (21), we hopefully obtain a vector for "watch" 👀 that is close to "look" and not to "clock". Analogously in (22), a vector for "mouse" should be close to "cursor" and not to "rat". The operations we define are parametrized in the degree of reliance they put on the expectations of the lexicon. To elucidate to which degree different ways of combining the lexicon and expectations, as well as the two alone, capture word interpretation, we implement the framework using three English LMs. For each LM, we deploy the framework on the CoInCo corpus (Kremer et al., 2014) – introduced in the previous chapter – and test the computed interpretations in the lexical substitution task; that is, to retrieve and rank appropriate paraphrases for a target word occurrence.

Our operationalization of the interaction between expected and lexical information bears strong ties to accounts in Linguistics and Cognitive Science. In particular, it connects to pragmatic theories taking expectations to guide the process of interpretation of expressions. Within the Gricean tradition, expectations of rational language use play a

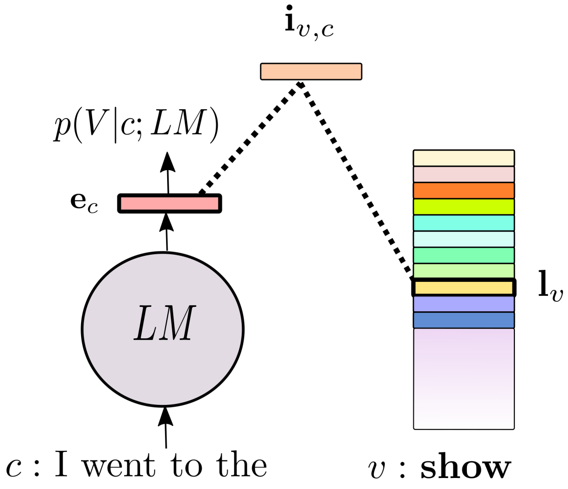


Figure 4.1: The context of utterance c is processed by the LM to yield an expectation $\mathbf{e}_c$, which is used to compute a distribution over the vocabulary given the context, $p(V \mid c; LM)$. The context-invariant word embedding of a word v, $\mathbf{l}_v$, represents lexical information. Both information sources are combined to yield a word's contextualized interpretation $\mathbf{i}_{v,c}$.

central role (Grice, 1975; Goodman and Frank, 2016) with listeners and speakers reasoning about each other's perspective, including during word interpretation (Wilson and Carston, 2007). Our approach is also related to accounts of word meaning access which take both expectations and lexical information to drive the activation of word information, by making more frequent and more contextually salient meanings more activated during comprehension (Duffy et al., 2001; Rodd, 2020). Finally, our framework shows links to accounts highlighting the role of predictability during processing (Levy, 2008; Clark, 2013; Kuperberg and Jaeger, 2016).

In the chapter, I address the following research questions:

RQ1. Can we use computational data-driven methods to obtain representations of expected content? In particular, representations that allow formalizing their combination with lexical content.

While representing the lexicon via word embeddings (or generally distributional representations) is not a novel technique, that of representing word-level expectations in their same space using LMs is, at least to the best of our knowledge. Typically, one focuses on the output probability distribution over words, which encodes relevant information, but not in a format compatible with that of lexical representations. As discussed and shown in Chapter 3, LMs – due to their training objective – accumulate word-level expectations about content in their internal representations. One of the intermediate activations of a LM (in certain conditions) can be considered a representation of expectations in the same space of word embeddings. This representation can be extracted directly from a pre-trained LM without the need of introducing any ad hoc modifications to the architecture, or of training an auxiliary model (as we did in Chapter 3). Once obtained the two representations of lexical and expected content, we can combine them using formal (vector) operations.

RQ2. To which extent the representations of the lexicon and expectations alone can explain word interpretation?

We expect both the lexicon and expectations alone to account for interpretation to some degree. This can be assessed by evaluating the quality of the representations to the extent to which they match human judgments of interpretations. On the one hand, the lexicon provides underspecified information, with a bias towards the most frequent senses. We can therefore expect this to be a useful pointer already in a good amount of cases, where the interpretation of the word is a frequent one. On the other hand, expectations have an important contextual bias in that they privilege content that seems relevant in a situation. Yet, they are based on an underlying uncertainty over the target word and thus may be at least partially off-track. We foresee the combination between lexical and expected content to outdo any of them alone, by combining the context-sensitivity of expectations with word-specific information.

RQ3. To which degree should a model of word interpretation rely on the lexicon or expectations to best explain word interpretation judgments?

Because of their complementary properties, a combination of lexical information and contextual expectations is expected to account for interpretation to a large extent. We instantiate our framework in models of interpretation adopting different degrees of reliance on the two sources of information and evaluate their resulting representations. It seems plausible that the reliance on lexical and expected information that best performs would not be extremely asymmetric, but rather one that substantially benefits from the two. Moreover, when using LMs of better quality – for instance, bigger and trained on more data –, expectations may acquire a more prominent role during interpretation, due to constituting a more reliable proxy. While we can foresee these patterns of behavior to better explain interpretation when kept constant across datapoints, it may be that interpretation is even better accounted for if the reliance on expectations and the lexicon dynamically vary on a case-to-case basis. We can explore this by looking at the variation of the per-datum optimal degree of reliance.

4.2 Modeling the Interaction between Expectations and the Lexicon

4.2.1 Representing Expectations and the Lexicon

To introduce the methods used in this framework, I again go through the way a LM process a text to compute an output word prediction, focusing on steps that justify why we can take certain representations as proxies for contextual expectations and the lexicon, respectively. I focus on aspects that are shared across models abstracting from specific architectural choices in which they may differ.

A neural LM is trained to output a probability distribution over the vocabulary V given a context (Figure 4.1). Given the linguistic context of the word at position $t - c_t$, deploying the LM gives us $P(V \mid c_t)$. We refer to a generic word type as v. As previously described, to process the text and compute a prediction, each word is encoded as a vector – a word embedding – which once learned during training remains static across contexts.[1] In practice, word embeddings are the row-vectors of an input matrix W_i: W_i has dimensionality $n \times |V|$, where n is the size of the embedding (a hyperparameter) and $|V|$ is the size of the vocabulary.

Lexicon. We can take word embeddings (rows in W_i) as the representations of lexical information, as we also did in Chapter 3. Following the same notation, we refer to the lexical representation of a word type v – its word embedding – as $\mathbf{l}_v$. This way of representing the lexicon adheres to distributional semantics: The similarity between words is measured by the geometric proximity of their representations (e.g., cosine), and is determined by their alignment in distribution (words that more often appear in similar contexts are closer). Indirectly, however, such similarity is a reflection of shared features between words (e.g., semantic or morpho-syntactic). As a result of the abstraction over a diverse corpus, these static embeddings may encode information relevant to different word senses, leading to underspecification. More frequent senses of a word will be more salient in the representation. This lexical information, therefore, provides an initial pointer to the information that is typically associated with the word.

How can we instead obtain representations of expectations from a LM? Concretely, we want a high-dimensional representation reflecting the type of content expected in a context, without having access to the word that would express it. This needs to be in the same space as the lexical representation to ease their combination through vector operations. Chapter 3 provided evidence that the predictive hidden states of a LM – those

[1] While this is the standard case for a word-level language model, not in all models each word is represented through an embedding: the input may instead be split into subword units (similar comment in Footnote 4 in Chapter 2).

used to predict a word given its context – encode expectations about the type of content that the target word expresses: with the aid of diagnostic models, we could retrieve a word representation that was reflecting, in terms of distance to word embeddings, those expectations. These representations thus satisfy the aforementioned requirements. However, the method that was used to isolate the relevant information in hidden states and map it to the desired space relied on a supervised auxiliary task, consequently requiring much initial work to obtain the representations. Besides, we cannot rule out that some shortcomings of our training data or setup may impede the extraction of all the relevant information in a hidden state (see discussion in 3.5.2). In principle, if an option, it would be desirable to extract representations of expectations directly from the LM without having to define and implement this machinery. As I explain next, this is actually not needed, as at least a group of LMs already have internal representations that satisfy the requirements to represent expectations.

Let us zoom in on the way a LM typically computes its word predictions. The embeddings of the words in c are processed to yield intermediate representations within the hidden layers of LM. Crucially, if we aim to represent expectations formed based on context only, we need to consider the case of word predictions where the target word is not accessed, using the same setups used to derive predictive hidden states (Section 3.2.2; e.g., in BERT substituting the word with `[MASK]`). Going through the depth of the network, the processing of the context c results in an activation vector **u** of size m such that passed through a linear mapping, involving an output matrix W_o, yields the output scores over the vocabulary.[2] The softmax function is applied to the output scores to obtain the final distribution $p(V \mid c_t)$: the higher the output score the higher the word probability.

$$\mathbf{u}_t = \text{LM}([\mathbf{l}_0, \mathbf{l}_1...]) \text{ where } [l_0, l_1...] = c_t \tag{4.1}$$

$$\mathbf{o}_t = \mathbf{u}_t W_o + \mathbf{b}_o \tag{4.2}$$

$$P(V \mid c_t) = \text{softmax}(\mathbf{o}_t) \tag{4.3}$$

To enable the computations, W_o has dimensionality $|V| \times m$. Depending on the LM architecture, $\mathbf{u}_t$ may be the last hidden state or a subsequent transformation of it.

We consider LM architectures – henceforth, **tied LMs** – where W_o is set to be the matrix transpose of W_i, meaning that the weights of the input and output matrices are shared (Inan et al., 2017; Press and Wolf, 2017):

$$\text{tied LM} \iff W_o = W_i^T \tag{4.4}$$

To enable this, one needs to set $n = m$, making the size of the word embeddings the same as that of representations **u** used to calculate the output predictions. Through this

[2] **b** is often not added; i.e., $\mathbf{b} = \overrightarrow{0}$. We can think of the **b** as a vector of learned word-specific constants (its size is $|V|$) which, re-weights the output scores, increasing or decreasing the probability of a word, independently of the context.

technique – a standard architectural choice in LMs – only the parameters of W_i have to be learned, therefore reducing the overall LM size (since the size of the vocabulary is often around 30-50K, the cut is considerable). Weight-sharing is possible because W_i and W_o share the size of one dimension, corresponding to the vocabulary size ($n \times |V|; |V| \times m$). Like W_i, W_o also learns one representation per word, in this case as its column-vectors. On top of the reduction of parameters, tying the input and the output matrices can therefore be motivated also by the benefits of sharing the word information that is learned both when representing its contribution (at the input) and when predicting it (at the output). This technique was empirically shown to enhance the quality of a LM, both in terms of perplexity in the language modeling task and the learned word embeddings (predicting word similarity judgments; Inan et al. 2017; Press and Wolf 2017; Gulordava et al. 2018a).

Let us now look at what the constraint in Eq. 4.4 implies for the information encoded in the representation **u**. We can effectively rewrite the multiplication between **u** and $W_o = W_i^t$ (Eq. 4.2) as the vector of dot products between each word embedding (column vectors in W_o) and **u**.

$$\mathbf{u}_t W_i^T = [\mathbf{u}_t \cdot \mathbf{l}_v | v \in V] \tag{4.5}$$

In the context of Eq. 4.2-4.3, the probability score assigned to a word v – $P(V = v \mid c_t)$ – therefore depends on the magnitude of the dot product between **u** and its word embedding $\mathbf{l}_v$ (determining the output score). During training, maximizing the probability of a word entails maximizing the dot product between the representation at **u** and the word's embedding. We can think of dot product as a measure of proximity in the high-dimensional space, or similarity, between two vectors (cosine is simply the dot product between the two vectors after normalization). Hence, a tied LM is optimized so that the similarity relations of **u** to word embeddings reflect the degree to which each word is expected.

Expectations. We take **u** to be a representation of contextual expectations about a word and henceforth refer to it as $\mathbf{e}_c$. This can be seen as a lower-dimensionality representation of the information encoded in the output distribution over the vocabulary: $P(V|c)$. How likely a word is encoded in terms of how similar the word's representation (**l**) is to $\mathbf{e}_c$.[3] The position of $\mathbf{e}_c$ in the word space can be seen not only as encoding which words are expected but also generally what type of content: if words referring to, e.g., animals are expected, $\mathbf{e}_c$ will be closer to nouns used to refer to animals. Since these

[3] The same argument – can be made – and therefore the framework implemented – for a non-tied LM if we take l_v to be the column-vector corresponding to v in W_o, instead of the row-vector of the shared input-output matrix. However, learning a unique representation for use at both the input and the output was found to lead to better representations, as well as output predictions (Press and Wolf, 2017; Gulordava et al., 2018a).

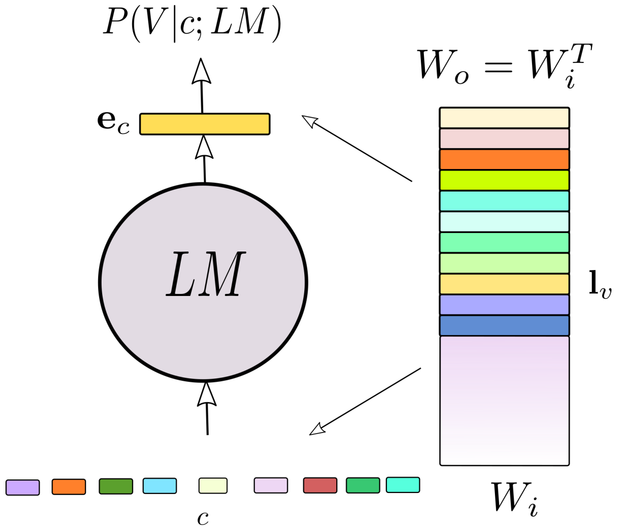


Figure 4.2: A schematic visualization of a tied language model, where the weights of the matrix used to represent the input words (input matrix; W_i) are shared with those of the matrix used to compute the output probabilities (output matrix; W_o).

are expectations computed based on context, they will reflect content that appears to be relevant in the situation; however, as the target word was not seen as input, $\mathbf{e}_c$ harbors uncertainty about its identity and associated content, and may therefore be misleading in some aspects.

These representations of expectations are based on the context as fed to the LM. If the context is restricted to the previous words, we can interpret these expectations as anticipation of information. If the context involves information that also follows the target word, we intend expectations in a broader sense as the type of information about a word contributed by the co-text of an expression – taken away the role of the word itself.

As mentioned earlier, what $\mathbf{u}$ – and therefore $\mathbf{e}_c$ – is within a tied LM may depend on its architecture. To better represent and therefore isolate the representation of contextual expectations certain architectural choices are preferable to promote a better division of labor in the LM. For instance, if $\mathbf{u}$ is the last hidden state in an RNN/LSTM (Section 2.2.1), this will not only constitute a pointer to the words expected by the LM, from which to compute probabilities but also, because of recurrent connections, will have to contain the information that is relevant to carry over to the adjacent timestep. This double function of the hidden state may be detrimental not only to the representation of contextual expectations, which would be mixed up with other information from the context but also to the quality of the LM. Gulordava et al. (2018a) empirically showed that the quality of LSTM LMs improves when in their architecture the representation of information relevant to the current prediction (i.e., expectations) is decoupled from the representation of the text processed so far, to be passed to the following timestep; that is, the hidden state. Even simply applying a linear mapping to the last hidden state leads to improvements in the language modeling task and the word embeddings. This can be explained in terms of the intermediate transformation enabling a division of labor, by extracting from the hidden state the information that is relevant for the current word prediction. Following these results, in our experiments we focus on tied LMs with an intervening transformation between the last hidden state and the calculation of output scores: $\mathbf{u}$ – and therefore $\mathbf{e}_c$ – is the result of applying a transformation to the last hidden state.

4.2.2 Combining Expectations and Lexical Information

For a target word occurrence – v in context c –, we now can obtain two representations, $\mathbf{l}_v$ and $\mathbf{e}_c$, respectively encoding the lexical information about the word and contextual expectations formed about its content. These representations have the same dimensionality, and based on the structure and training of the LM, they encode information across their dimensions in a corresponding way. We can now define operations to combine these two representations, or else put, sources of information, to yield a representation of the interpretation of a word. We refer to this output representation – a function of $\mathbf{e}_c$ and $\mathbf{l}_v$ – as $\mathbf{i}_{v,c}$.[4]

$$\mathbf{i}_{v,c} = f(\mathbf{e}_c, \mathbf{l}_v) \tag{4.6}$$

Generically, this implements the idea that the interpretation of a word is guided by: 1) the generic lexical information associated with a word, 2) expectations formed in the current context. We define and test two operations (versions of f; 4.6), both acting as a way of combining information from the two vectors and parametrized by the degree to which expectations and the lexicon contribute to the resulting interpretation. This allows us to investigate the optimal degree to which to attend to these sources of information when interpreting words.

One operation – *avg* – corresponds to the **weighted average** of the two vectors.

$$\textbf{\textit{avg}}: \qquad \mathbf{i}_{v,c} = \alpha\, \mathbf{l}_v + (1 - \alpha)\, \mathbf{e}_c \tag{4.7}$$

Intuitively, this operation blends expected and lexical information, highlighting what is common between the two: the part of the lexical information that is relevant to the context. The value of each dimension of $\mathbf{i}_{v,c}$ is computed as the weighted average of the respective values in $\mathbf{l}_v$ and $\mathbf{e}_c$: dimensions with high (or low) values for both vectors will also be high (or low) in the resulting vector. Depending on the parameter $\alpha \in [0; 1]$, the

[4] The vectors are normalized before applying these operations.

resulting vector is influenced more by one information source than the other: if $\alpha = 0$, the interpretation solely relies on contextual expectations; conversely, if $\alpha = 1$, the interpretation only corresponds to lexical information. When $0 < \alpha < 1$, both lexical and expected information are taken into account, with perfect symmetry when $\alpha = 0.5$.

The other operation, **delta rule**, reduces the distance between the expectations and lexical information, to a degree that is constrained by an α parameter:

$$\textbf{\textit{delta}}: \qquad \mathbf{i}_{v,c} = \mathbf{e_c} - \alpha \nabla D_{\mathbf{e}_c}, \text{ where } \quad D = 1 - cos(\mathbf{e}_c, \mathbf{l}_v) \tag{4.8}$$

This is achieved through gradient descent: $\mathbf{e}_c$ is shifted in the direction of the negative gradient of D with respect to $\mathbf{e}_c$ ($\nabla D_{\mathbf{e}_c}$). α regulates how close expectations are pulled towards lexical information, with an analogous role of the learning rate when gradient descent is applied for machine learning. If $\alpha = 0$, the interpretation coincides with contextual expectations: $\mathbf{i}_{v,c} = \mathbf{e}_c$. As α approaches infinity, the contribution of $\mathbf{l}_v$ grows and that of $\mathbf{e}_c$ shrinks. Differently from *avg*, we do not a priori know which α value marks where the lexicon is more taken into account than expectations and vice-versa (in *avg*, 0.5). However, to facilitate the understanding of the results, we can empirically establish the value where the interpretation (on average) becomes closer to one or the other (see, e.g., Figure 4.3).

We can think of the *delta* operation as a form of "expectation revision", which adapts formed expectations to information encoded in the actual input. During interpretation, we may want to reduce the uncertainty or potential error in the expectations, by getting closer to what is encoded in the lexical information; at the same time, however, we may want to keep some of the context-sensitivity of expectations. In principle the *delta* operation could also be designed in other direction (switching $\mathbf{e}_c$ and $\mathbf{l}_v$ in Eq. 4.8), adapting instead lexical information to expectations. Despite the conceptual difference, we expect this alternative operation to lead to analogous results.

These vector operations have the potential of accounting for both the conventional and innovative aspects of lexical ambiguity. Information that is often associated with a word is highly activated in the lexical vector, and further enhanced if in line with expectations: this facilitates the retrieval of familiar senses of a word. However, the output of the interpretation process is not limited to a pre-defined set of senses, so contextual nuances can be accommodated. But also where the interpretation ends up in the high-dimensional space is not bounded, so one could even represent novel usages of a word that deviate much from what it typically conveys (e.g., figurative language).

4.3 Experiments

4.3.1 Language Models

The framework described in the previous section can in principle be applied to any LM provided that certain conditions are met: that the input and output matrices are tied, and preferably the presence of a transformation between the last hidden state and the computation of the output probabilities (Gulordava et al., 2018a). For our experiments, we instantiate our framework using three pre-trained English LMs, comparing the results obtained with each. All models are bidirectional for analogous reasons to those presented in the previous chapter. In particular, as evaluation data, we consider annotations collected in an off-line task (after finishing reading a passage of text) where both the left and right context of a word were accessible and thus could affect the interpretation. However, in principle, our framework can also be instantiated using unidirectional LMs, taking into account only one side of a context. The approach would be recommendable, for instance, when focusing on online processing effects, or if strictly focusing on expectations as anticipation.

The first model we consider is a variation of the **biLSTM** presented in Chapter 3. This is identical in terms of architecture and training (for details, see Section 3.2.1), except for a few aspects. Concretely, the model employs weight-sharing between the input and output matrices, with an intermediate non-linear transformation between the last hidden state and the output computation. Recall that the last hidden state used to predict the word at t is obtained processing the left and right context of a word up to position $t-1$ and $t+1$, respectively, through the two unidirectional LSTMs.

$$\mathbf{u}_t = \tanh\big((\overrightarrow{\mathbf{h}}^{L}_{t-1} + \overleftarrow{\mathbf{h}}^{L}_{t+1})W_u + \mathbf{b}_u\big) \tag{4.9}$$

$$\mathbf{o}_t = \text{softmax}(\mathbf{u}_t W_i^T) \tag{4.10}$$

The other LMs we use are the two versions of the transformer-based BERT model released by Devlin et al. (2019) – **BERT-base** and **BERT-large**. In the previous chapter, we employed only the former model, as its lower number of layers simplified our analysis of word information across its depth. We here also use the bigger BERT-large (340M parameters as opposed to 110M). BERT-large has 24 hidden layers with 16 attention heads, and an embedding and hidden state size of 1024 units (see 3.2.1 for the comparison to BERT-base). Except for these differences, the two LMs share all other aspects of the architecture and training. In particular, the weights of the input and output matrices are tied, and a non-linear transformation is applied to the last hidden state to compute probabilities (Eq 3.2-3.3 in Chapter 3). To predict a word, it is substituted in the input by the `[MASK]` token.

To run our framework, given a word occurrence, we deploy the pre-trained LM on the text to extract representations: for a word v in context c, we extract $\mathbf{e}_c$ and $\mathbf{l}_v$. For all

LMs, $\mathbf{e}_c$ is the result of non-linear transformation applied to the last hidden state used to predict the word v (the *predictive* state; Section 3.2.2); $\mathbf{l}_v$ is the word embedding of v. We then compute the interpretation $\mathbf{i}_{v,c}$ yielded through *avg* or *delta* with the chosen α parameter.

4.3.2 Data and Evaluation

We aim to assess the overall ability of our framework to account for word interpretation, and to compare different ways of combining lexical and expected information (through different degrees of reliance on either of them; i.e., the α parameter). For this evaluation, we make use of the Concepts in Context (CoInCo) dataset (Kremer et al., 2014), an English corpus annotated for lexical substitution (**LexSub**). The dataset was already presented and used for experiments in Chapter 3. It contains 15.5K content words in context (at most 3 sentences) annotated with crowd-sourced paraphrases. Contextual substitutes of a word reflect the inferred interpretation by the annotators. Such judgments are given as an off-line task: they are not given at the time when the word is processed but, in principle, after the speaker finished reading the entire passage of text (including context after the target word). This dataset enables us to test our framework: (1) on a large scale and a relatively natural distribution of words and their interpretations, and (2) on both sharp and nuanced differences among word usages, without having to assume a predefined sense ontology.

We evaluate vector representations of words in the degree to which their position in the high-dimensional space, and therefore the information encoded, reflect the way humans interpreted that word occurrence. We consider both the substitute retrieval and ranking tasks, following the same method described in Section 3.3.3. Given a representation, we use cosine similarity scores between this and a word's embedding to estimate its plausibility as a substitute. For ranking, we order all substitutes of a word type across the dataset by cosine to the evaluated vector and compare the ranking to the datapoint's gold standard – GAP (Thater et al., 2009). For retrieval, we take the 10 highest ranking word lemmas in terms of cosine and measure their overlap with substitutes provided by annotators – Recall-10 (McCarthy and Navigli, 2007). Both GAP and Recall-10 factor in the number of annotators that provided a substitute, with more weight given to paraphrases produced by more annotators; for both scores, the higher the better.

The CoInCo data comes with a split in data tobe used for development and testing – *dev*/*test* (10K/5K word occurrences, further filtered by coverage of the vocabulary of the LM).[5] As opposed to the experiments in Chapter 3, we do not need to select a portion of the *dev* data to use for training, as we here do not rely on supervision. We can instead use the full *dev* set to study the effect of modulating α on accounting for word interpretation. For each LM and operation, we evaluate the data at 10 α-values, in the range $[0, 1]$ for

[5] See footnote 8 in Chapter 3 for details about the coverage of the LMs.

	expected	lexical	avg	delta
biLSTM (avg: α=0.5; delta: α=0.9)				
GAP	0.45	0.45	**0.48**	**0.48**
Recall-10	0.13	0.34	**0.38**	**0.38**
BERT-base (avg: α=0.4; delta: α=0.6)				
GAP	0.51	0.45	**0.52**	**0.52**
Recall-10	0.31	0.40	**0.48**	**0.48**
BERT-large (avg: α=0.3; delta: α=0.6)				
GAP	0.52	0.46	**0.54**	0.53
Recall-10	0.34	0.43	0.49	**0.51**

Table 4.1: Results on *test* data in ranking (GAP) and retrieving (Recall-10) substitutes, with best constant α.

avg and $[0, 3]$ for *delta*. We then assess:

- the α value, for each LM and operation, yielding the best mean performances on the LexSub tasks when α is kept constant across data points (we refer to this as **constant** α). This value is then used for evaluation on the *test* data.
- for each datapoint (i.e., word occurrence annotated with substitutes), the α value yielding the best performance on the LexSub tasks relative to that case (we refer to this as **optimal** α). We then compute the mean performance in the LexSub tasks across the dataset when considering, for each datapoint, the result obtained with its optimal α value.

In both cases, we take an α-value to perform *best* if yielding the highest sum of Recall-10 and GAP scores, under the assumption that a good representation of interpretation should maximize both. Comparing performances using constant and optimal α values is indicative of whether α better works as a datum-dependent or -independent parameter, or – else put – if word interpretation is better modeled considering a context-dependent reliance on expectations or the lexicon.

4.3.3 Results

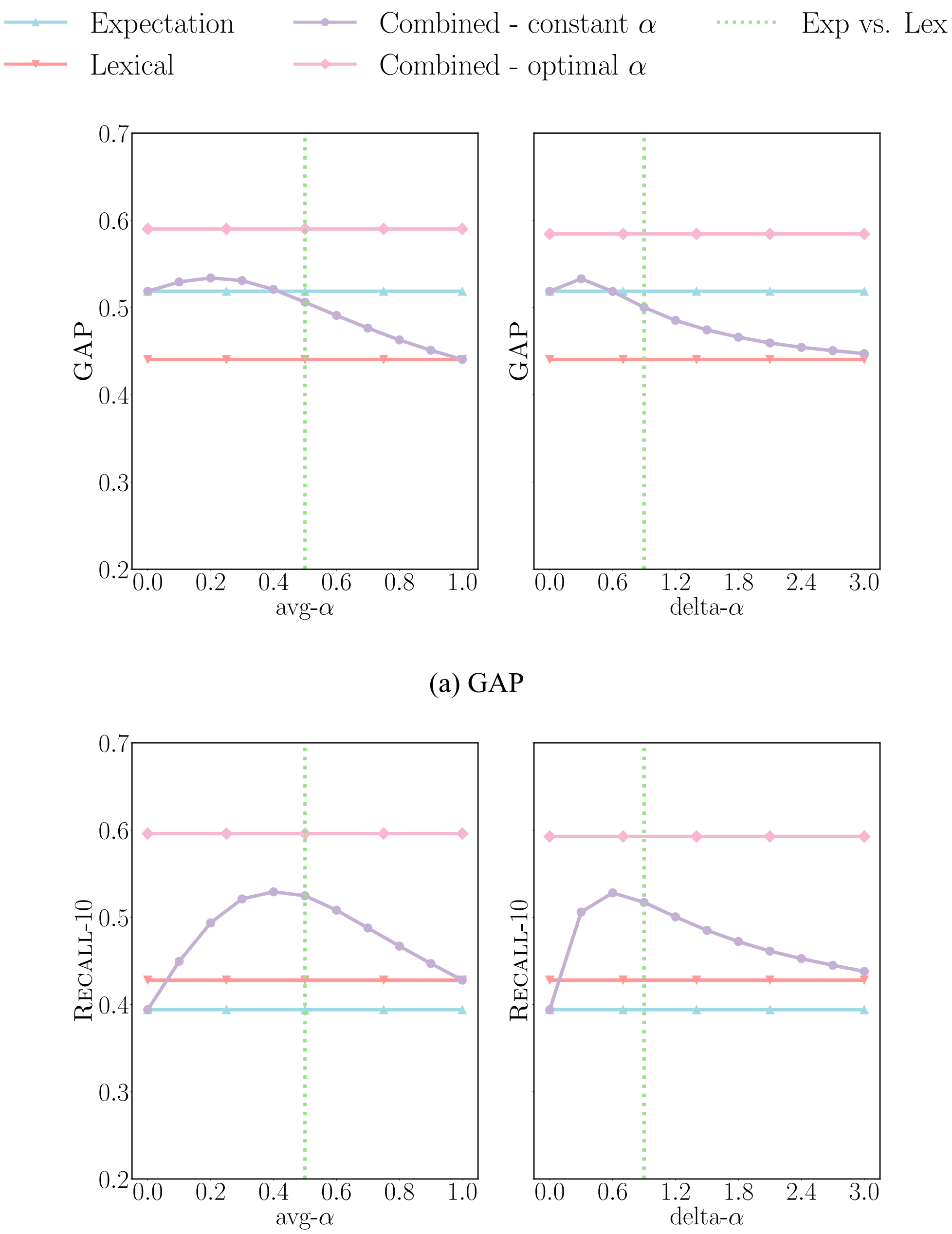
Table 4.1 reports the results of the evaluation on *test* data for each LM and operation. We compare the scores when focusing only on expected information ($\alpha = 0$), only on lexical information ($\alpha = 1$), or to both to some degree modulated by a constant α, estimated based on *dev* data. We focus first on expectations and lexical expectations alone. Since expectations lie on uncertainty about the target word, they often fail to have substitutes as the closest words (Recall-10); lexical information performs best, despite being static across word occurrences, as its neighbors are words generically related to the most frequent senses and therefore often provide good guesses. However, when ranking word-specific candidates (GAP) the context-sensitivity of expectations is advantageous. Though expectations don't frequently have substitutes as neighbors, they are quite successful at ranking candidate substitutes in terms of how relevant they are to the context.

Nevertheless, as we expected, it is when we combine lexical and expected information that we obtain the best results in both Recall-10 and GAP. This result is found using both the *avg* and *delta* operations, and all LMs. The fact that both Recall-10 and GAP increase speaks to the generally improved quality of the resulting combination of lexical and expected information. The results obtained are competitive with recent NLP approaches to lexical substitution (e.g., Garí Soler et al. 2019b; Zhou et al. 2019; Arefyev et al. 2020).[6]

In terms of overall results, performances with *avg* and *delta* tend not to differ (except for a bit with BERT-large) and are even identical in the biLSTM. Through additional analysis, we find that their results tend to align to a large extent not only at the level of the average performances across the dataset, but also that of specific datapoints: There are very strong correlations of the performance scores achieved on each case with *avg* and *delta* operations (e.g., for BERT-large, Spearman's $\rho = .9$ for Recall-10; =.97 for GAP). Analogously, if we look at the overlap in the top 10 closest lemmas of $\mathbf{i}_{v,c}$ (candidate substitutes as evaluated in Recall- 10) for the two operations, we find that on average the vast majority is shared (e.g., for BERT-large 95%). These results suggest that the two operations, *avg* and *delta*, implement analogous mechanisms of a combination of lexical and expected information, resulting in similar output behavior.

Table 4.1 shows that the LMs differ in the preferred constant α, but generally all favor an intermediate value that is focusing neither on expectations nor on the lexicon to an extreme degree. On *dev* data, we inspect more closely how the behavior changes as α varies. This is displayed in Figure 4.3, showing the results on *dev* LexSub data using different α values kept constant across datapoints. We focus on BERT-large, which achieved the best results, but analogous trends are found when looking at the other LMs (see Appendix B.1 for the full results).Starting from expectations ($\alpha = 0$), increasing the reliance on lexical information progressively improves the results: incorporating such information allows for a better interpretation, presumably to fill the uncertainty gap that

[6] It is unclear whether all approaches computed the evaluation metrics in the same way we did. For instance, some works build on the evaluation procedure (and code) from Melamud et al. (2015) where GAP is computed considering as candidates to rank only substitutes associated with that word type and the current part of speech (e.g., "show" as a verb). This substantially simplifies the task, by removing the challenge of syntactic category ambiguity. Consequently, I do not go focus much on the comparison to previous work, especially considering the more linguistic-oriented goals of our work.



(a) GAP

(b) Recall-10

Figure 4.3: Results on *dev* data for BERT. Dotted lines mark whether a given α yields a $\mathbf{i}_{v,c}$ that is closer to $\mathbf{e}_c$ (to the line's left) than to $\mathbf{l}_v$ (to its right).

<table>
<tr><td colspan="2">(1)... The flavor is relatively mild, with a fresh, sour <b>note</b>
substitutes: <i>taste, aftertaste, flavor, hint, jolt, suggestion, tinge, tone, undertone</i></td></tr>
<tr><td>exp</td><td><i>taste, flavor, texture, appearance, smell</i></td></tr>
<tr><td>lex</td><td><i>notice, letter, mark, tone, message</i></td></tr>
<tr><td>optimal α: 0.1</td><td><i>taste, flavor, feeling, texture, appearance</i></td></tr>
<tr><td colspan="2">(2) ... Karnes had his own Jeep, and went to the <b>beach</b> and to evening movies
substitutes: <i>shore, coast, dock, oceanfront, sand</i></td></tr>
<tr><td>exp</td><td><i>movie, theater, park, library, office</i></td></tr>
<tr><td>lex</td><td><i>shoreline, coast, coastline, shore, coastal</i></td></tr>
<tr><td>optimal α: 1</td><td>=lex</td></tr>
<tr><td colspan="2">(3) ... He took another <b>shot</b> of vodka.
substitutes: <i>drink, glass, cup, hit, jigger, measure, sip, swig</i></td></tr>
<tr><td>exp</td><td><i>drink, sip, hit, pull, swallow</i></td></tr>
<tr><td>lex</td><td><i>film, shooter, cut, fire, stab</i></td></tr>
<tr><td>optimal α: 0.1</td><td><i>drink, hit, sip, pull, swallow</i></td></tr>
<tr><td colspan="2">(4) ... and I <b>hope</b> to see you again!
substitutes: <i>want, wish, desire, aspire, crave, pray</i></td></tr>
<tr><td>exp</td><td><i>want, promise, plan, expect, wish</i></td></tr>
<tr><td>lex</td><td><i>hopeful, expectation, promise, optimistic, fear</i></td></tr>
<tr><td>optimal α: 0.4</td><td><i>want, promise, wish, expect, desire</i></td></tr>
<tr><td colspan="2">(5) ... a huge staircase that seemed to go up for more <b>stories</b>
substitutes: <i>floor, level, fable, flight, landing, plateau, tale</i></td></tr>
<tr><td>exp</td><td><i>flight, level, floor, step, attempt</i></td></tr>
<tr><td>lex</td><td><i>tale, narrative, storylines, storyteller, myth</i></td></tr>
<tr><td>optimal α: 0.2</td><td><i>flight, floor, level, tale, step</i></td></tr>
<tr><td colspan="2">(6) ... He accepts his role with a club and he’s a team <b>guy</b>
substitutes: <i>player, man, dude, member, person</i></td></tr>
<tr><td>exp</td><td><i>player, man, person, leader, girl</i></td></tr>
<tr><td>lex</td><td><i>dude, kid, girl, man, boy</i></td></tr>
<tr><td>optimal α: 0</td><td>=exp</td></tr>
</table>

Table 4.2: Examples of predicted substitutes (5 closest lemmas) for target word occurrence (in bold) for the expectations, lexical representations, and its optimal combination.

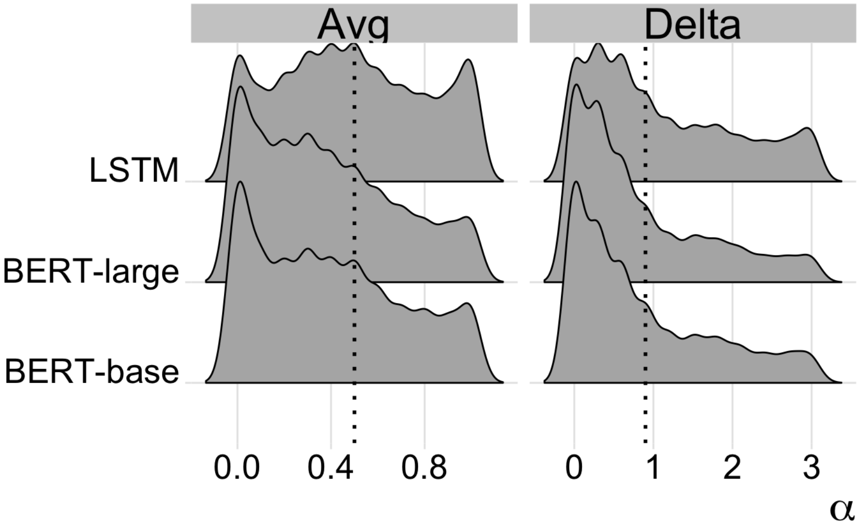


Figure 4.4: Distribution of optimal per-datum α using each LM. Dotted lines mark whether a given α yields a representation that is closer to $\mathbf{e}_c$ (to the line's left) than to $\mathbf{l}_v$ (to its right).

expectations have about the target word. After reaching a peak where the resulting interpretation outdoes both expectations or the lexicon alone, the results, degrade to then reach the same performance as that of lexical information (as its reliance on it is increased). This indicates that looking too much at lexical information ends up being misleading; this is likely due to the ambiguity or bias towards non-relevant senses this can introduce. The best behavior is then found at an intermediate point that allows benefiting from the context-sensitivity of expectations as well as the generic information about what the word tends to usually encode (i.e., lexical). As we saw in Table 4.1, where this intermediate peak is found (best constant α) varies across models. While the biLSTM favors an equal reliance on expectations and the lexicon (e.g., for *avg*, $\alpha = .5$), with BERT models we observe a slight preference for expectations (e.g., for *avg*, $\alpha < .5$). This is signaled in Figure 4.3 by marking the α-value that result in interpretations that are closer to $\mathbf{e}_c$ than to $\mathbf{l}_v$. These differences across LMs may be explained to their different quality in terms of language modeling itself (dependent on aspects like the size of a model or its amount of training data), which should impact the quality of the expectations formed: the better the model is at word prediction, the better its representations of expectations are.

Given these results, it is plausible that the best constant α may decrease as a measure of the LM's quality, with expectations taking a more prominent role as they become more reliable. It is however unlikely that even a "perfect" LM – completely simulating human-like expectations – would favor $\alpha = 0$, where lexical information is completely ignored. Due to the pervasive uncertainty that underlies communication, knowing that a specific word was uttered and the information this word is typically associated with is critical for comprehension, even if the context provides much information about what it may convey. We, however, also note that providing both the left and right context of a word may largely reduce the uncertainty over expectations. Instantiating our framework with a unidirectional model may lead to different observations.

Lastly, while a certain α value for a LM led to the optimal behavior when kept constant across datapoints, it is not necessarily the parameter that allows to better model each word occurrence: certain cases may be better modeled with increased or decreased focus on the lexicon (and consequently decreased or increased focus on expectations). We study this by exploring the behavior under different α values for each datapoint on *dev* data. Figure 4.3 compares the performance on LexSub tasks that are obtained with constant α values and when instead picking, for every datapoint, its optimal α-value (Figure 4.4 shows the distribution of optimal α across datapoints). The latter method outperforms any constant α, indicating that the balance between the optimal contribution of $\mathbf{e}_c$ and $\mathbf{l}_v$ varies across words and contexts. This result suggests that the extent expectations or the lexicon are to be trusted is a contextual matter.

Qualitatively inspecting examples such as those in Table 4.2 provide some insights as to the factors driving this variation. In cases such as (2), expectations are not off-track but lie on much uncertainty: likely some place would be mentioned but with many diverse options, among which “beach”, the correct one, may not rank high. Its lexical information is therefore crucial, and, since “beach” is not a particularly ambiguous word, can be trusted fully (i.e., assigned maximum weight for interpretation). By contrast, in (1), interpreting “note” in the context of describing a flavor requires to focus much on expectations, as the lexical information has a bias to the non-relevant verb reading or the 📝 sense. In other cases, like (4), both expectations and the lexicon are to be taken into account: the former accounts for syntactic cues leading to expect a verb, modulating the ambiguous lexical content of “hope” towards the contextually appropriate reading. What these examples suggest is that what mainly contributes to increasing the bias to either source of information is the accuracy of expectations (i.e., how well they are approximate what is actually conveyed), on the one hand, and the degree of dominance of senses as encoded in the lexical information (i.e., how strongly information relevant to the current interpretation is encoded), on the other.

4.4 Summary and Discussion

In this chapter, I described a computational framework to model word interpretation and the results of its deployment on an English corpus to model off-line interpretation judgments (through contextual substitutes). I first proceed to discuss our findings in the context of linguistic research, and then discuss the advantages of our framework and ways in which it could be extended.

4.4.1 The Interplay between Expectations and the Lexicon in Interpretation

How a word is interpreted depends on an interaction between the type of information it is usually associated with and the context it is embedded in – linguistic or extra-linguistic. This basic assumption underlies the core idea of what it means for an addressee to interpret a word: to retrieve the content that the word comes to convey with an utterance, considering both the type of information the word typically conveys and what would be more in line with the current context. In this chapter, I presented a computational framework of lexical ambiguity resolution, where interpretations arise as a combination of these two sources of information: *lexical information* and *contextual expectations*. We proposed to encode these through representations from pre-trained LMs, which were in the same high-dimensional space and therefore could be combined through vector operations. Within the framework, models with different degrees of reliance on either source of information can be implemented, in order to explore the interplay between the lexicon and expectations that underlies the way lexical ambiguities are resolved.

We instantiated our framework using three English LMs, and used it to analyze a large-scale corpus. In particular, we computed representations of interpretations and evaluated them in the extent to which their word similarity relations – and therefore the position in the high-dimensional space – reflected the way humans interpreted the words, as emerging from substitutes provided. Our results indicate that both lexical and expected word information, taken separately, codify information relevant to interpretation, but when combined they perform best. The sweet spot where their combination is optimal is an intermediate point where both play a non-marginal role. For instance, with the operation *avg*, the best constant α was for all LMs in the range 0.3-0.5, whereas 0.5 corresponds to equal reliance on the lexicon and expectations, and 0 and 1 to ignore one or the other. LMs that are bigger and trained on more data (BERT vs. biLSTM) were associated with increased reliance on expectations, surpassing that on the lexicon (with *avg*, $\alpha < 0.5$): when the expectations formed by the LM are of better quality, they more reliably contribute to interpretation.

That expectations need to be put more weight on than the lexicon to account for word interpretation can be explained as follows. Expectations circumscribe the area of the conceptual space that a speaker is expected to convey according to the context. This information has an important context-sensitivity, but harbors uncertainty over the actual word to interpret and the content this tends to encode. Under the assumption that a large amount of information is predictable in language, expectations will however often be on the right track (i.e., predicting relevant information), and thus a smaller – yet crucial – contribution of lexical information (low α parameter) may already be effective.

The behavior described above is the one found to perform best when kept constant across word occurrences (constant α). However, it does not have to be the disambiguation strategy that works optimally for each case. We found that the optimal weight put on the two sources of information varies much across cases (Figure 4.4), and leads to a better account of word interpretation than keeping the behavior constant. This suggests that the reliance on expectations and the lexicon gets regulated in context, and raises questions regarding the factors driving this variation. From a qualitative inspection, what seems to mostly drive the division of labor between expectations and the lexicon is (1) the accuracy of expectations, and (2) the dominance of a certain sense as encoded in the lexical information. The first effect is compatible with accounts taking the role of expected information to be facilitatory for processing (Kuperberg and Jaeger, 2016), such as surprisal theory (Hale, 2001; Levy, 2008) and predictive coding (Clark, 2013): the more expectations deviate from information compatible with the actual input, the more laborious the revision process to adjust to the target word will be. The effect of dominance in the encoded lexical information is in line with the approaches taking the frequency of a sense to affect the information activated during comprehension (Rayner and Frazier, 1989; Duffy et al., 2001), or assuming that some senses are more *literal* than others (Asher and Lascarides, 2003; Wilson and Sperber, 2006). Frequent interpretations are easier to retrieve and are the ones taken by default unless the context strongly supports information that deviates from them.

More analyses would be beneficial to further understand the interplay between expectations and the lexicon during interpretation. It would be interesting to assess whether different mechanisms are to be posited for different relations among possible senses of a word (e.g., different morpho-syntactic categories, related or unrelated senses), or rather based on a word's or context's property (e.g., sense dominance, or contextual uncertainty). This could be studied focusing on data targeting specific subcases of lexical ambiguity resolution (from syntactic disambiguation to metaphor comprehension), as opposed to the current analyses where we focused on the framework's explanatory breadth at scale, considering trends in a large and diverse corpus. Modeling could also offer insights into the division of labor between expectations and the lexicon; in particular, by extending the framework's structure to allow for the parameter of reliance on the two inputs (α) to be dynamically adjusted to the context.

The framework presented resonates with a series of proposals on ambiguity resolution and processing, providing a methodology to instantiate and test some of their assumptions through computational modeling. While some choices may be considered limitations or elements of divergences (discussed more below), the methodology I presented represents a step forward to empirically ground mechanisms that are hypothesized to characterize ambiguity resolution. In particular, our approach is in line with the relevance-theoretic approach to lexical pragmatics. This takes word interpretations to be derived considering lexically encoded information and contextual expectations of relevance (Wilson and Carston, 2007). A modulation process adjusts the concept encoded by a word (through processes like narrowing or loosening) until reaching an interpretation that meets contextual expectations of relevance. Analogously, in our framework, the interpretation is yielded by identifying an intermediate point between the lexical and expected information, highlighting their shared aspects. Differently from relevance theory, our procedure does not consider an iterative process stopping when a suitable interpretation is found (to minimize cognitive cost), but rather expectations and the lexicon are combined in one step. Within pragmatics, our proposal also shows links to the Rational Speech Act framework (Goodman and Frank, 2016): Expressions are associated with a literal interpretation (typically, a distribution over meanings or features). This is then used to compute a new interpretation (pragmatic or non-literal) taking into account expectations about rational language use from the side of the interlocutor (e.g., what would be likely to convey in context; e.g., Kao et al. (2014)). The literal interpretation plays an analogous role to lexical information in our framework, as it gets refined considering the addressee's contextual expectations.

Approaches that are also similar in spirit to our framework are accounts of word meaning access (Rayner and Frazier, 1989; Duffy et al., 1988; Rodd, 2020). These may differ in the assumed time-course of word information activation, but overall share the idea that the information activated relevant to a word's interpretation is dependent on (1) information associated to its diversity of senses, with an advantage of dominant senses, and (2) information supported by the context. As a result of the strength of activation, an interpretation is selected (and the non-relevant information is suppressed). Akin to this, in our framework, the lexical information makes available familiar senses of a word with a bias towards the most frequent ones, while the expectations point to information that is relevant to the context. The vector operations strengthen the activation of information that is both lexically and contextually congruent.

While word meaning access approaches tend to focus on rather sharp sense distinctions, pragmatic approaches tend to look at more nuanced cases (e.g., polysemy), including more idiosyncratic usages (e.g., "Ironing is the new *yoga*"; Wilson and Carston 2007). Our proposed framework may therefore constitute a bridge to put together these perspectives and provide a unifying account of word interpretation.

4.4.2 Methodological Considerations

The method we used to implement our model of interpretation involved representations from pre-trained LMs as proxies for contextual expectations and the lexicon, and the use of vector operations to combine them. It did not rely on supervision as to how words should be interpreted, but only leverages the knowledge formed by a LM about word-specific information (the word embeddings) and what is expected in a context (word-level predictions). An important benefit of this approach, when comparing to other accounts of lexical ambiguity (Small et al., 2013; Falkum and Vicente, 2015), is that it can be flexibly applied to different combinations of words and contexts, and therefore enables large-scale evaluations of the posited interpretation mechanism.

Computational models of ambiguity resolution often need to rely on annotations or intuitions about what is, for instance, the lexical information associated with a word (its potential meanings) or what cues in the context can bias an interpretation. Crucially, this constrains both the coverage and diversity of cases that a computational model can be tested on. For instance, in the computational model of figurative interpretations of words by Kao et al. (2014), the focus is on metaphorical interpretations of words referring to animal categories. A word's literal sense is encoded as a vector of elicited features (e.g., *scary* for "shark"), and the linguistic context the word is embedded in is as simple as, e.g., "He is a *shark*", as a response to being asked how a certain person is.

Earlier models of ambiguity resolution in Computational Linguistics have benefitted from the use of distributed representations learned entirely from data. In particular, a research branch focused on how distributional semantic representations of words, underspecified in nature, can be contextualized to express the contextual contribution of the word (Erk, 2010). The disambiguation process these works implement can be described as co-composition (Pustejovsky, 1995; Asher et al., 2017), in that the representations of the words in the sentential context are combined to modulate the lexical information (Erk and Padó, 2008; Mitchell and Lapata, 2008; Thater et al., 2011). Such approaches either apply a shallow processing of the context (considering bags of words) or stipulate an elaborate interaction with syntax; either way, accounting for context beyond the sentence boundaries, and therefore discourse factors, proved difficult within such framework (Bernardi et al., 2015). Deep learning models supposed a big step forward for an automatic processing of linguistic sequences in that they learn their way of encoding its content by combining the input words. Instead of focusing on representations reflecting how the model itself interpreted a word, we extract the representations about the information that the model expects based on the context. This provided a way of operationalizing the influence that a discourse context has on interpretation, which can be then combined with that of lexical information.

The methodology we used therefore has a series of advantages. Yet, in future research more analyses of the adequacy of the representations used in our framework could be beneficial. We have built on existing evidence that embeddings derived from a word's distribution across contexts approximate lexical information to large extent, as shown in tasks ranging from similarity relatedness scores to predict patterns of brain activation (see Lenci 2018 for an overview). However, few studies focused on the impact of ambiguity on distributional representations; in particular, whether their underspecification is a good parallel to the way humans encode lexical information, or, for instance, the word embeddings overrepresent dominant senses. As for the representations of expectations, evaluations of LMs tend to focus on perplexity (looking at the probability assigned to the word actually uttered in a corpus) but not on, for instance, modeling distributions from a cloze task (e.g., what words humans would predict to fill a gap in a sequence). The latter type of evaluation, especially if carried out on a large scale, can help to assess whether the word-level expectations formed by LMs correctly reflect the uncertainty humans have.

We can envision various ways in which in future work the current framework could be implemented differently than in our experiments, or modified to accommodate assumptions. For instance, many studies on ambiguity resolution, and in particular those focusing on its time-course, focus solely on the role of the previous context for disambiguation, while we also incorporated the following context (using bidirectional LMs). The framework is, however, not bounded to this choice and, depending on the pretrained LM used to obtain representations, can be made to use only the prior linguistic context (with a unidirectional model), or even to incorporate the extra-linguistic one (e.g, through an image; Lu et al. 2019). Moreover, some approaches reject the idea that lexical information is represented in a unique representation, but rather senses are separately stored (Klein and Murphy, 2001). One could modify the current framework to operate with multi-prototype distributional representations (Schütze, 1998; Reisinger and Mooney, 2010), where each sense of a word gets its vector representation.

Finally, one could use an analogous approach to that used in this chapter to obtain proxies for expectations at different levels, for instance as an explicit probability distribution over the potential interpretations of an expression. This is the approach I take in Chapter 6 where a deep learning model is trained to compute a probability distribution over the entities an expression could be referring to, on the basis of its context. This allows us to study the role of contextual expectations on reference production.

Chapter 5

SYNTACTIC AMBIGUITY RESOLUTION IN NEURAL LANGUAGE MODELS

In this chapter, I study the way neural language models (LMs) process and resolve ambiguities by quantifying their degree of uncertainty over a set of candidate interpretations. In particular, I focus on the resolution of **temporary syntactic ambiguities**. A temporary ambiguity occurs when, during the incremental processing of a sentence, no contextual cues are available so far to determine a unique interpretation; in the case of syntax, multiple parses are simultaneously tenable (Frazier, 1978). As the sentence unfolds, its resulting structure clarifies the intended interpretation, deeming the initial ambiguity only temporary. The following sentences exemplify the case of the NP/S ambiguity:

(23) a) *The fans knew the singer* since 2008.

b) *The fans knew the singer* was going to release a new album.

The portion in italics is temporarily ambiguous as "singer" could act both as the direct object of the main verb (23a) or as the embedded subject in an upcoming subordinate clause (23b) – NP and S readings, respectively. When a LM process such an input does it remain uncertain over the two interpretations, or does it have a default bias towards one? Does the LM correctly adapts its behavior when disambiguating cues are instead given?

I address these questions by estimating the underlying probability distribution over interpretations in a LM, both when the input is ambiguous and when instead it contains cues to the intended interpretation. This allows us to observe the degree of bias of the LM for each interpretation when the input is in principle compatible with multiple ones – its default preferences under ambiguity–, as well as the way the behavior changes as the LM receives evidence for the intended interpretation – its context-sensitivity.

In Chapter 3, to uncover the way lexical ambiguities are resolved, we looked at the information encoded in the internal representations of LMs. I here introduce and apply a behavioral analysis method, which focuses instead on the probabilistic output of LMs. Concretely, we look at text generation as a window into a LM's processing of an unfolding sentence: by looking into how the model would complete an input we estimate its degree of preference for its candidate interpretations.

5.1 Approach and Research Questions

Temporary syntactic ambiguities like NP/S (e.g., (23)) have received much attention in psycholinguistics to study mechanisms in human sentence comprehension. In particular, they proved useful to better understand the incremental parsing process: for instance, whether when processing a sentence, only one or a subset of the possible parses are considered (Frazier and Fodor, 1978), or all in parallel weighted by probability (Hale, 2001). Moreover, it has been investigated whether the default preferences over the interpretation of a temporary ambiguity solely depend on syntactic principles, or are also affected by lexical factors (MacDonald et al., 1994; Garnsey et al., 1997). The way an interpretation is revised in cases of initial misinterpretations has also been a subject of study (Pritchett, 1988; Grodner et al., 2003; Christianson et al., 2001). Analogously, temporary ambiguities are also useful to study the underlying syntactic processing in LMs. These deep learning models, despite not receiving any explicit signal on how to parse sentences during training, were shown to exhibit much awareness to syntactic structure, both at the level of their predictions (e.g, Linzen et al. 2016; Gulordava et al. 2018b; Wilcox et al. 2018) and the information encoded in the internal representations (e.g., Giulianelli et al. 2018; Hewitt and Manning 2019).

Within the study of temporary ambiguities, much research focused on explaining and modeling *garden-path* effects (Bever, 1970) – a high cognitive cost at disambiguation taken to signal a preference for the alternative analysis. For instance, in (23b), the garden-path effect would occur at "was" if the NP interpretation was initially preferred (since "was" introduces the S reading). Estimates, like word surprisal, from unidirectional LMs (processing text incrementally) have been used to model such effects in terms of predictability (Van Schijndel and Linzen, 2018, 2021). But also one can test whether LMs themselves exhibit some kind of garden-path effects (Futrell et al., 2019). These results indicated that, as the LM processes the text incrementally, it considers a certain interpretation of the input more likely, leading to high surprisal when this turns out to be ultimately incorrect. Yet, the degree of imbalance to which a LM expects that interpretation has not been directly quantified. As a result, it is not clear whether on this type of ambiguities LMs tend to (1) select an interpretation, due to some default bias (only to be revised at need), or (2) keep multiple options open until disambiguating evidence is provided. This chapter attempts to clarify this by providing a general methodology to qualify the implicit distribution over interpretations of an input entertained by a LM.

We probe the degree of syntactic uncertainty that LMs have when processing temporary ambiguities, using **generation** as an analysis tool. Generating text using a LM's output probabilities can be seen as a way of instantiating a LM's expectations over an unfolding sentence. As (23) show, the completion of a temporarily ambiguous sentence portion clarifies the intended interpretation. We can therefore inspect how the LM interprets an input by using its next-word probabilities to complete the sentence. Building on this idea, we apply the following methodology to estimate the implicit distribution over interpretations of the input: Given a target text as input (containing or not disambiguating cues as to its interpretation), we generate a set of sentence completions of an input by repeatedly applying stochastic decoding (e.g., sampling words from the output distribution). A sufficiently large set of completions can be seen as exemplifying what the LM expects moving forward in the sentence, by exploring the LM's probabilistic output. The proportion of completions that are consistent with a certain reading of the prompt – derived through an automatic classification – is taken to indicate the degree that this reading is expected by the LM. Therefore, we use the relative frequency of that reading in the sample to estimate its probability.

The idea of analyzing the text generated by a LM to understand its underlying dynamics has not been explored much in previous work. Futrell et al. (2018) studied whether LMs are aware of obligatory syntactic events by testing them on a relative clause completion task (e.g., after "The authors who the editor...", the LM should complete both the relative and the main clause). From each input, 9 completions were generated via stochastic decoding and then manually annotated to establish their correctness. Our methodology is analogous but we consider a much larger amount of completions to estimate the underlying distribution of a model over the input's parse. Van Schijndel and Linzen (2021) analyzed syntactic predictions of LMs after temporary ambiguous sentence portions, looking at the next-word probabilities and grouping words by part of speech. Though they focus on just one word, their approach is connected to ours in that the expectations of the LM over an unfolding sentence are analyzed grouping its predictions – in their case at the word level – based on syntactic information. We instead classify a LM's sentence completions. Another relevant work is that by Wei et al. (2018), which looked at the syntactic properties of the text generated by machine translation systems. Analogously to our method, they rely on the use of an external syntactic parser in order to analyze the generated text.

We consider three types of temporary syntactic ambiguities in English (NP/S, NP/Z, Noun/Verb; I present these in Section 5.2); for each type, we derive prompts from sentences drawn from psycholinguistic experiments (Grodner et al., 2003; Frazier and Rayner, 1987). We compare the LM's uncertainty on ambiguous prompts (e.g., underlined region in 23), as well as unambiguous prompts that vary in the number and location of disambiguating cues (before and/or after the element involved in the ambiguity). To classify each generated sentence based on the interpretation it is associated with, we use a set of rules based on the dependency labels predicted by a syntactic parser. The English LMs we analyze in this study are the LSTM model released by Gulordava et al. (2018b) and the transformer GPT2 (Radford et al., 2019). Both models were trained to carry out unidirectional language modeling, processing the text from left to right and basing their word predictions only on the previous context. This is motivated by the current focus on the way ambiguities are incrementally resolved as the LM processes an unfolding sentence and incorporates disambiguating cues. From a methodological perspective, also, only this class of LMs can be used to generate text.

In spite of the focus on temporary syntactic ambiguities, the results of these analyses can be informative also about the general ambiguity resolution strategy of LMs. In Chapter 3 we studied this focusing on lexical ambiguities by looking at the way the word information is processed across the depth of the network. To do so, we considered the case where the LM could access both the left and right context of the work, assuming (with some degree of simplification) that all disambiguating material was provided to allow the LM to infer the correct interpretation. We here instead focus on studying the behavior of the LM in the absence or presence of disambiguating cues, in terms of the implicitly maintained distribution over interpretations. Temporary syntactic ambiguities offer a controlled setup to study this. First, these syntactic ambiguities come with a known discrete set of candidate interpretations, as well as clear contextual cues which strictly constrain the interpretation of the sentence (i.e., excluding the alternatives). By contrast, for semantic ambiguities, it would be more complex to list all candidate interpretations as well as to identify the specific cues in a context that constrain them. Finally, due to their temporary aspect, these ambiguities offer a way of observing the time-course of ambiguity resolution in a unidirectional model – processing text from left to right in English – by looking into how the interpretation of the LM changes as this processes more information.

With these analyses, we thus address the following research questions about the processing of ambiguities in LMs:

RQ1 When the input processed by a LM is ambiguous (i.e., it can be interpreted in multiple ways), does the LM a priori favor one interpretation, consequently assigning it (nearly) all the probability mass? Or does it track and consider plausible multiple interpretations?

LMs assign probabilities to words, while, as shown, keeping track of the syntactic information in a sentence to a large extent. They can therefore be compared to a probabilistic parser implicitly evaluating the likelihood of certain parses of a sentence (Futrell et al., 2019). When processing an ambiguous input, where multiple interpretations are possible, the LM could exhibit different behaviors in terms of the degree of preference for each interpretation. Concretely, it could display none or low degree of syntactic uncertainty, if assigning all or most of the probability to one interpretation; or instead consider viable to a relatively large extent multiple interpretations, until disambiguating cues are provided.

RQ2 Does an interaction between syntactic and lexical factors modulate the probability of a syntactic interpretation entertained by a LM?

The items we use for each temporary ambiguity have the same sentence structure but vary in their lexical items. If the LM's default resolution of these ambiguities (without disambiguating cues) depended solely on syntactic preferences, we should expect no variation in terms of probabilities of interpretations across items that differently realize the same ambiguity. Conversely, if we observe variation, this indicates that the LM's preferences are modulated in a context-dependent fashion. These results will allow us to understand whether the LM's behavior over syntactic ambiguities is therefore the result of abstract syntactic preferences (e.g., minimal attachment; Frazier 1978), or an interaction of syntactic and lexical factors (Garnsey et al., 1997).

RQ3 How responsive is a language model to disambiguating cues in the input that reveal the correct interpretation?

When disambiguating cues are available in the input, the interpretation of the sentence is clear and a LM would behave correctly if it assigned all the probability mass to this one, discarding the other alternatives. The lack of this behavior would suggest disambiguation issues, and therefore room for improvement for the context-sensitivity of the LM during their processing of an input. In particular, we compare how a LM reacts to different types of evidence for disambiguation: cues that appear before and/or after the ambiguous region.

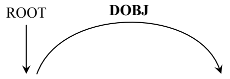
5.2 Temporary Syntactic Ambiguities

This section describes the types of temporary ambiguity and the materials used in this study. These ambiguities are structural, and therefore affect the overall interpretation of the sentence and resulting parse. However, they can be distinguished based on the syntactic role of a certain word in the sentence, which we here refer to as the **locus of the ambiguity**.

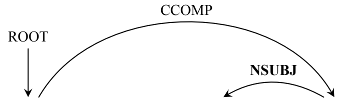
5.2.1 The NP/S Ambiguity

The NP/S ambiguity was already briefly introduced in the previous section as an example of syntactic ambiguity. I here present it in more detail. The sentence portion in (24) is compatible with the main verb "understood" taking either a noun phrase (NP) or a sentential (S) complement. This is reflected by the syntactic role of "contract" – the locus of the NP/S ambiguity – which could act as the direct object of the main verb (labeled as root; 24a), or as the embedded subject in an upcoming subordinate clause (24b).

(24) The employees understood the contract ...



a) The employees understood the **contract** *well.* → **NP**



b) The employees understood the **contract** would *be changed very soon.* → **S**

An equivalent of (24b) without temporary ambiguity can be obtained by adding the complementizer "that" after the main verb, preempting the NP reading:

(25) The employees understood *that* the contract would be changed very soon. → **S**

For these experiments, we use the 20 NP/S sentence pairs from Grodner et al. (2003) – *unmodified* versions.[1] Each pair consists of a temporarily ambiguous sentence and its unambiguous counterpart, both of which eventually have an S interpretation. The first shares the basic structure of (24b); the other of (25). The items are built to create an initial bias for an NP interpretation: First, the main verb appeared more often with a direct object complement than a sentential one (estimated through the PennTreebank corpus). Second, the locus of ambiguity is a plausible direct object of the main verb. Nevertheless, an S interpretation is still possible, and we aim to quantify whether a LM considers this as likely, or instead assigns all its probability to the NP interpretation.

From each sentence pair (ambiguous and unambiguous sentence), we derive four types of **prompt**; i.e., the sentence portion which will be passed to the LM as input for generation, and therefore whose processing we analyze. Examples of prompts are shown in Table 5.1. No cue prompts correspond to the sentence portions with temporary ambiguity. The other prompts contain at least one disambiguating cue, before or after the locus of ambiguity: "that" is the **pre-locus cue**, while the **post-locus cue** is the word immediately after the locus of ambiguity. This is a finite verb that introduces the verb of the sentential complement. The specific word acting as post-locus cue varies across items.

[1] Grodner et al. (2003) considered two variants of sentences, with or without material between the locus of ambiguity and the post-locus cue (*modified* and *unmodified*, respectively). We use the unmodified sentence pairs for both NP/S and NP/Z.

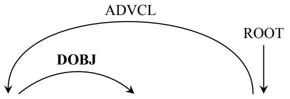
Prompt type	
no cue	The employees understood the contract
post-locus cue	The employees understood the contract **would**
pre-locus cue	The employees understood **that** the contract
pre&post-locus cues	The employees understood **that** the contract **would**

Table 5.1: Examples of prompt types for NP/S; locus of ambiguity underlined, disambiguating cues in bold.

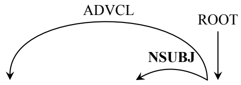
5.2.2 The NP/Z Ambiguity

In (26), the verb "left" in the subordinate clause can be parsed as taking either a noun phrase complement (NP) or none (zero complement; Z). The locus of ambiguity is "party", which can be the direct object of "left" (26a) or subject of the upcoming main verb (26b).

(26) Even though the band left the party ...



a) Even though the band left the **party** *I stayed.* → **NP**



b) Even though the band left the **party** *went on for another hour.* → **Z**

The unambiguous version of (26b) explicitly marks the boundaries between the subordinate and main clauses, through the insertion of a comma:

(27) Even though the band left**,** the party went on for another hour. → **Z**

We use the 20 *unmodified* (see footnote 1) NP/Z sentence pairs from Grodner et al. (2003). Both sentences in each pair ultimately had the Z interpretation, following the structure of (26b) and (27), respectively. Analogously for NP/S, the ambiguous NP/Z sentences are constructed to create an initial bias for the NP interpretation. The verb in the subordinate clause appears more often in transitive than intransitive constructions, and the locus of ambiguity is a plausible direct object.

From a sentence pair, we derive prompts following analogous criteria described for NP/S ambiguity (cropping the sentence before and after the locus of ambiguity, respectively). In this case, the pre-locus cue is the comma, while the post-locus cue – varying across items – is a finite verb intended to act as the main verb of the sentence (examples in Table 5.2).

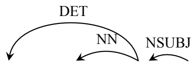
Prompt type	
no cue	Even though the band left the party
post-locus cue	Even though the band left the party **went**
pre-locus cue	Even though the band left**,** the party **went**
pre&post-locus cues	Even though the band left**,** the party **went**

Table 5.2: Examples of prompt types for NP/Z; disambiguating cues in bold, locus of ambiguity underlined

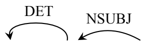
5.2.3 The Noun/Verb Ambiguity

The last ambiguity we investigate involves words that have both a noun and verb reading. This case is part of the general phenomenon of lexical ambiguity, which we already looked into in the previous chapters. We here look into this further, in particular focusing on those cases where the lexical ambiguity leads to a temporary structural ambiguity. This is, for instance, the case of sentences like (28): if "suit", the locus of this ambiguity, is a noun, "pants" acts as its modifier; otherwise it acts as its subject.

(28) Mary thinks that the pants suit ...



a) Mary thinks that the pants **suit** *is pretty*. → **Noun**



b) Mary thinks that the pants **suit** *me well.* → **Verb**

The temporary ambiguity in (28) can be preempted by replacing "the" with a determiner that forces an interpretation through number agreement:

(29) a) Mary thinks that *this* pants suit is pretty. → **Noun**
b) Mary thinks that *these* pants suit me well. → **Verb**

We study this type of ambiguity using the data from Experiments 1 and 2 of Frazier and Rayner (1987). Differently from the other ambiguities' data, for each temporary ambiguity, two sentence pairs are provided, with Noun and Verb interpretations, respectively ((28a) and (28c) for Noun, and (28a) and (28b) for Verb). A minority of cases employ disambiguating cues other than agreement, for instance additionally or alternatively changing the word before the locus of ambiguity. In order to consider a coherent set of datapoints, we discard such cases, therefore focusing solely on the role of agreement for disambiguation. This leaves us with 26 sentence pairs (out of 32) each for Noun and Verb interpretations, respectively. We obtain prompts from the pairs, treating the determiner as pre-locus cue in this case. As we have one pair of sentences for each

Prompt type	
no cue	Mary thinks that the pants suit
post-locus cue	
Verb	Mary thinks that the pants suit **me**
Noun	Mary thinks that the pants suit **is**
pre-locus cue	
Verb	Mary thinks that **these** pants suit
Noun	Mary thinks that **this** pants suit
pre&post-locus cues	
Verb	Mary thinks that **these** pants suit **me**
Noun	Mary thinks that **this** pants suit **is**

Table 5.3: Examples of prompt types for Noun/Verb; cues in bold, locus of ambiguity underlined

reading (Noun and Verb), for unambiguous prompt types (all but No cue), we derive two subtypes of prompts, as shown in Table 5.3. We can therefore inspect the behavior of a LM in both of the scenarios where the disambiguating cues lead to the Noun and Verb resolution of the ambiguity, respectively.

5.3 Methods

5.3.1 Language Models

We evaluate two English language models. Both models are trained on unidirectional, or *causal*, language modeling: for each word prediction, they solely take into account the previous linguistic context. As mentioned earlier, this is motivated by the focus on temporary ambiguities, where incrementality of processing plays a crucial role: a bidirectional model, accessing both the left and right context of a word, would be able to observe the entire sentence, thus at any point accessing cues that give away the correct interpretation. Moreover, we can only use this class of model for text generation (at each timestep, we need to generate the next token given the preceding ones). Therefore, the methodology we propose for this study can only be run on such models.

The first LM we evaluate is the **LSTM** released by Gulordava et al. (2018b). This LM has two hidden layers; the hidden and input embedding size is set to 650 units. Through recurrent connections between timesteps, the model processes the text from left to right. The LM was trained on an 80M-tokens Wikipedia corpus, with a vocabulary of 50K words. The other LM is the transformer-based **GPT2** in its *small* version (Radford et al., 2019).[2] This model builds on the architecture of Radford et al. (2018) and consists

[2] GPT2 is used through the Transformers library (Wolf et al., 2020). For text generation, we build on

of 12 hidden layers of transformer blocks with 12 attention heads each. The hidden and embedding size is set to 768 units. Masked self-attention is applied in order, for each timestep, not to attend to future positions in the input, and thus output a prediction only based on the previous context. The model was trained on the 40GB WebText corpus, using Byte Pair Encoding (Sennrich et al., 2016). It has a vocabulary of ≈50K items, comprising both words and subwords. GPT2, which surpasses the LSTM in both model and training corpus size, was shown to have remarkable text generation abilities. We can therefore expect this LM to generate more fluent text.

5.3.2 Estimating Syntactic Uncertainty

Text Generation

From a prompt we generate a completion through **stochastic decoding**: at every iteration of the generation algorithm, a word is generated sampling from the LM's output distribution over the vocabulary. Concretely, the LM processes an input and outputs a probability distribution over the next token (Eq. 5.1); from this, a word is sampled (Eq. 5.2) and incorporated in the input. The process is then repeated to generate the next tokens.

$$P(X_{i+1}|x_{1:i}) = LM(x_{1:i}) \tag{5.1}$$

$$x_{i+1} \backsim P(X_{i+1}|x_{1:i}) \tag{5.2}$$

To obtain the sentence completion of a prompt, we iterate the generation step for a fixed number of tokens, then crop the text to automatically identified sentence boundaries.[3] More details are provided in Appendix C.1.

There exist variants of stochastic decoding which modify the LM's output distribution before sampling, such as truncated sampling or the use of temperature (Holtzman et al., 2019). Though these were found to improve the quality of the generated text, applying such techniques alters the LM's output behavior. For this reason, for the main experiments, stochastic decoding is applied with no such modifications in order to estimate the LM's uncertainty before any intervention that could modify it. However, in Section 5.5.1 we analyze how changes in the decoding strategy affect the text generated by the LM on ambiguous prompts.

Estimating the Probability of an Interpretation

The syntactic ambiguities presented in the previous section all consist of a scenario where the locus of ambiguity has two potential interpretations, which we here refer to

the available decoding functions, adapting them to also work on the LSTM.

[3] Using Spacy Sentencizer.

generically as i_1 or i_2. In principle, however, our method can be generalized to the case with more than two interpretations.

We aim to estimate the probability that the LM assigns to each interpretation after processing the prompt; that is, a Bernoulli distribution such that:

$$P(i_1|\text{prompt}) = 1 - P(i_2|\text{prompt}) \tag{5.3}$$

We derive an empirical estimate of this distribution by independently sampling a set of sentence completions of the prompt (C_{prompt}) and classifying them by the interpretation they imply of the locus of ambiguity. In our experiments, we generate 100 completions of each prompt, sampled with replacement (i.e., the same completion may be generated more than once).[4] The relative frequency of interpretations in the sample is then used to estimate their probabilities:

$$\hat{P}(i_1|\text{prompt}) = \frac{|\{c \in C_{\text{prompt}}|\text{interpretation}(c) = i_1\}|}{|C_{\text{prompt}}|} \tag{5.4}$$

This allows us to quantify the degree of preference of the LM for each interpretation of the prompt, and thus its uncertainty. For unambiguous prompts, the desired behavior is that the probability of the correct interpretation is 1, as the LM only generates completions that are consistent with it. In the presence of ambiguity (No cue prompts), the LM could distribute the probability mass across multiple interpretations. The degree to which this occurs reveals how uncertain the LM is about the interpretation of the input. If almost exclusively generating completions of one type, we can conclude that the LM has a strong default preference for that reading; otherwise, that the LM considers both as viable options, possibly to different degrees.

This operationalization thus enables us to inspect the question of whether the LM's behavior on syntactic ambiguities is driven by absolute or graded default preferences (RQ1), and whether such preferences vary across instances of an ambiguity suggesting an interaction between syntactic and lexical factors (RQ2). The former aspect is observable looking at the variation of the estimated probability values across items. Finally, by looking at the model's behavior on unambiguous prompts, we can check whether it uses the disambiguating cues to commit to the correct interpretation (RQ3).

Completion Classification

The method described above requires to classify completions based on the syntactic interpretation of the locus of ambiguity (e.g., for NP/S prompts, whether they are completed leading to an NP or an S interpretation.) This can be done through manual or

[4] In practice, the same completions were rarely sampled more than once. The average proportion of unique completions in a sample (pooling together all prompt types) is at least 98% for all ambiguity types and LMs. We report further results on the diversity of generated completions in Appendix C.2.

automatic classification. A manual classification is highly reliable, but less practical when a large set of sentences needs to be analyzed (in our case, 100 for each prompt). Moreover, annotations regarding the syntactic analysis of a sentence require linguistic expertise and therefore would be complex to collect, for instance, through crowdsourcing. By contrast, an automatic classification relies on the use of an external syntactic parser, thus making the process much more efficient. However, the predicted classification may introduce noise in case the parser itself incorrectly disambiguates the sentence. Even though state-of-the-art parsers are highly reliable, sentences with potential ambiguities may indeed be a source of potential errors.

As a compromise, we use automatic annotations and assess their quality by comparing them to manual annotations collected for a subset of sentences. This step acts as a sanity check to assess whether the automatic annotations tend to be reliable. For the automatic classification, we use the AllenNLP (Gardner et al., 2018) dependency parser, based on the model of Dozat and Manning (2017). This reaches a label attachment accuracy with predicted part-of-speech tags of 92.86%. Once a sentence is parsed, we use a set of rules, defined for each ambiguity type, to classify completions based on the predicted labels. It may occur that a completion cannot be traced back to either of the candidate interpretations. This tends to be due to some error of the parser, an ill-formed completion, or a correctly detected alternative interpretation. Since in practice, this is a rare scenario, for simplicity, such completions are discarded from the sample for the analysis. For each ambiguity type, I report the rules used to classify completions in its relevant section; more details are then presented in Appendix C.3.

For each type of ambiguity, a random sample of 80 sentences (20 × prompt type) generated by GPT2 was manually annotated. This is carried out by three trained linguists, each of whom reviews data from a different ambiguity type. The annotator chooses between the two candidate interpretations, but also *other* and *unclear* (if an alternative or no interpretation can be derived, respectively). Binary judgments on the syntactic well-formedness of the sentences are also collected (we do not consider the semantic plausibility of the sentence, due to the focus of our study on syntax). Since character and punctuation errors are frequent (e.g., a misspelled word, or the incorrect presence of a punctuation mark), annotators can specify when a sentence would count as grammatical without such errors. Overall, 66% of NP/S completions, 61% of NP/Z completions, and 66% of Noun/Verb completions are judged to be fully well-formed. If we ignore spelling and punctuation errors, these percentages increase to 75%, 74%, and 85%. Given that the presence of grammatical errors was relatively much more frequent than the use of the label *unclear*, we conclude that, at least to some extent, the suboptimal quality of a generated text does not limit the applicability of this method: the function of the locus of ambiguity is still typically distinguishable.

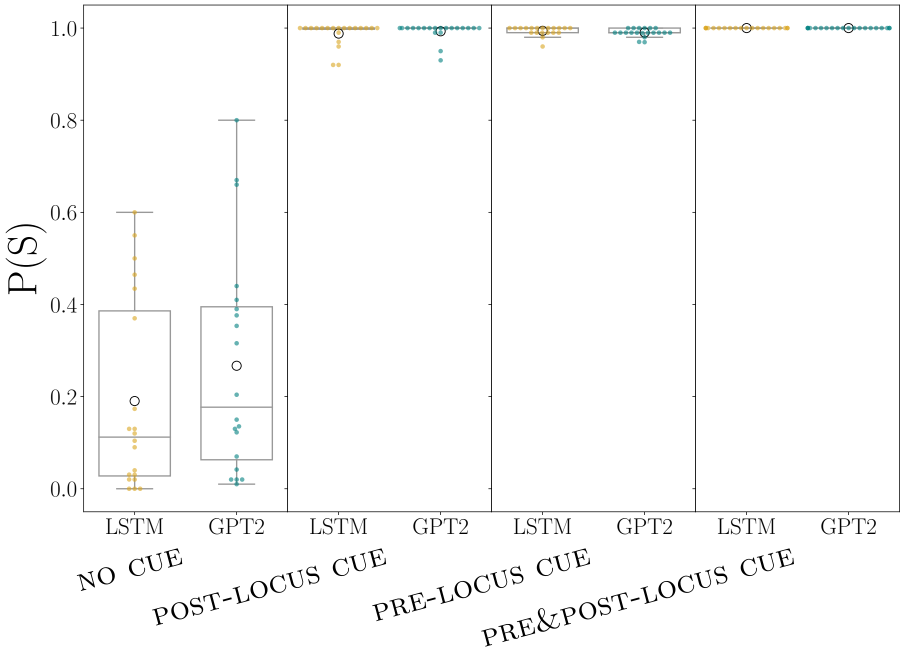
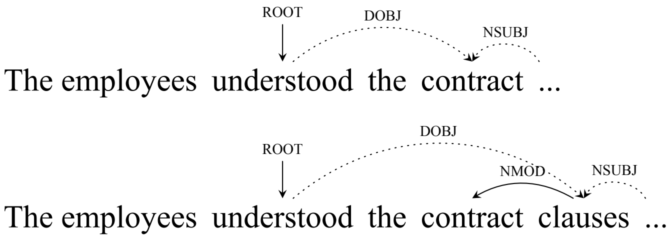


Figure 5.1: Distribution of $P(\mathrm{S})$ for each NP/S prompt type and LM. Each dot represents the value for a prompt item, as estimated considering its completions sample; the circle indicates the mean across items.

5.4 Results

5.4.1 The NP/S Ambiguity

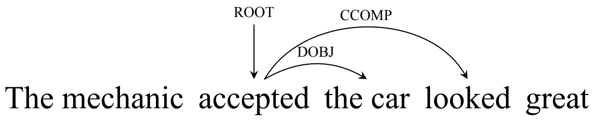
Classification. The NP and S interpretations can be distinguished through the syntactic role of the head of the noun phrase that contains the locus of ambiguity (direct object or subject leading to the NP and S interpretations, respectively; see the example of (24)): The locus of ambiguity can itself be the direct object or subject; or it can be part of a complex NP, where, for instance, it is the modifier of another noun.



The employees understood the contract ...

The employees understood the contract clauses ...

To classify completions based on this criterion, we then define a set of rules, based on the labels predicted by the dependency parser (reported in Appendix C.3). To reduce noise from parser errors, we define a heuristic that corrects the most typical type of misclassification (NP instead of S). This involves a failure to detect S cases, labeling the locus of ambiguity as a direct object when followed by a finite verb:



This parse is not only incorrect but also ungrammatical, as it leaves the verb after the locus without a subject. The classification is therefore corrected to NP.

If these rules do not identify the interpretation as either NP or S, the sentence is discarded from the completions; for both LMs, this is the case for 0.2% of the completions, across all prompt types. We validate the classification method in two ways. As a first check, we run it on the original sentences from which the prompts were derived, whose correct interpretation we know from the start: all of them are correctly classified. We then compare the collected manual annotations to the automatic ones, and find the latter ones to be reliable: There is a near-perfect agreement between the manual and automatic annotations (Cohen's k = .96). The annotator never used the label *unclear*.

Results. Based on the distribution of NP and S completions in the generated sentences, we compute $P(\text{S})$ for each prompt ($P(\text{NP}) = 1 - P(\text{S})$). The distribution across items for the different prompt types is visualized in Figure 5.1, where each dot represents the value of $P(\text{S})$ for a certain prompt, computed from its completions sample. Examples of completions for the different prompt types can be found in Table 5.4.

We focus first on the No Cue prompts, which are ambiguous between NP and S. Both LMs are often uncertain – to varying degrees – between the two interpretations of a prompt. With exceptions, they exhibit a preference for NP ($P(\text{S}) < .5$), though this preference is typically not absolute, as S completions are also generated (e.g., (1b) in Table 5.4). This indicates that, in the presence of the NP/S ambiguity, the LMs tend to consider multiple parses at the same time. In spite of the general preference for NP, $P(\text{S})$ values vary across items: syntactic uncertainty depends on the context, with some cases even favoring an S analysis. This indicates that lexical factors modulate the syntactic expectations of the LM, with different instances of the NP/S ambiguity (differing in their lexical items) giving rise to different default preferences. The two LMs exhibit the same trends, though GPT2 seems to favor more often the S interpretation than the LSTM.

The other prompt types all contain at least one cue disambiguating the sentence as S (before and/or after the locus of ambiguity): for the LM to behave correctly, it should generate only completions consistent with S, as NP would lead to an ungrammaticality. In line with this prediction, for all these conditions and LMs, $P(\text{S})$ is very close to 1. Considering that in No cue prompts the LMs tend to favor NP, this indicates that the LMs are responsive to the disambiguating cues and use them correctly to adapt their behavior. A qualitative inspection of the sentences supports this observation, as there is no

(1) The scientist proved the theory	
a) *through two experiments.* (NP)	b) *was correct.* (S)
(2) The tourists saw the palace was	
a) *on fire.* (S)	b) *under construction.* (S)
(3) The journalist confirmed that the story	
a) *is false.* (S)	b) *was being reported on his network.* (S)

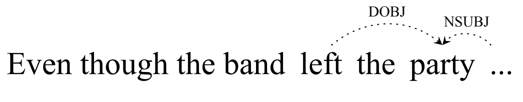
Table 5.4: Examples of completions generated by GPT2 for NP/S prompts.

evidence of disambiguation issues: the LMs correctly pursue the interpretation clarified by the disambiguating cue (e.g., (2-3) in Table 5.4). A minority of completions of unambiguous prompts are classified as NP. This occurs due to occasional misclassifications, but also ill-formed completions whose interpretation is unclear (e.g., "The employees understood that the contract."; completion underlined), or when NP is actually licensed despite the post-locus cue (e.g., "The army found the supplies saved by the French.").

Overall, we can conclude that the LMs tend to display some degree of syntactic uncertainty when processing NP/S ambiguities. The detected general preference for NP over S is compatible with results reported by other studies looking at the behavior of LMs on NP/S ambiguities (Van Schijndel and Linzen, 2018) and in particular their surprisal level when encountering an S analysis. Analogously, humans were found to exhibit garden-path effects over NP/S sentences indicating that they initially favor the NP interpretation (for instance, on these very same items, see the results by Grodner et al. 2003). In our results, we could also see that the degree of preference of a LM for the NP is usually non-absolute (the S interpretation is also tracked) and also vary from case to case, with even some cases favoring an S analysis. When disambiguating cues are given, the LMs correctly adapt to pick the intended interpretation.

5.4.2 The NP/Z Ambiguity

Classification. NP and Z interpretations can be distinguished following the same criterion of the NP/S ambiguity: based on the syntactic role (direct object or subject for NP and Z, respectively) of the locus of ambiguity.



Even though the band left the party ...

We thus employ the same set of rules we used for distinguishing NP and S cases. The rule correcting cases where a subject is labeled as direct object tend to be crucial for this classification, as the parser is prone to errors on NP/Z temporarily ambiguous sentences. A total of 0.6% and 1.4% of GPT2 and LSTM completions, respectively, cannot be identified as either NP or Z, and are thus discarded from the analysis. When run on the

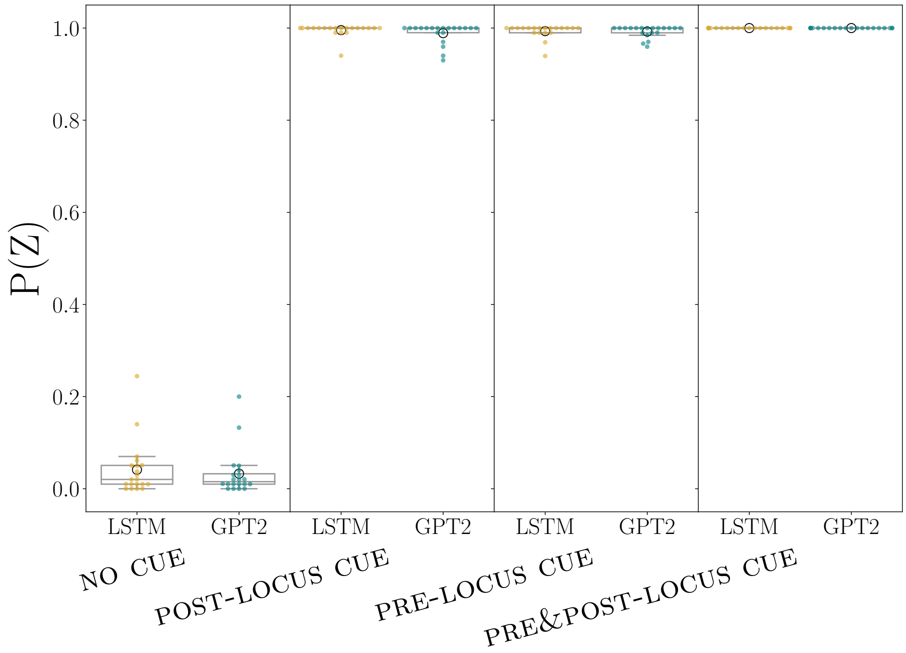


Figure 5.2: Distribution of $P(\mathrm{Z})$ for each NP/Z prompt type and LM. Each dot represents the value for a prompt item, as estimated considering its completions sample; the circle indicates the mean across items.

original sentence pairs, the classification always derives the correct analysis (Z). The agreement between the automatic and manual annotations is high (Cohen's $k = .86$); the divergences that occur are all due to the use of the label *unclear* by the annotator (4 sentences), an option which is not available to the parser.

Results. Figure 5.2 shows $P(\mathrm{Z})$ values ($P(\mathrm{NP}) = 1 - P(\mathrm{Z})$) for each prompt type. Examples of completions are reported in Table 5.5.

In the ambiguous No cue prompts, there is limited syntactic uncertainty: $P(\mathrm{Z})$ stays close to 0 (on average, .03 and .04 for GPT2 and LSTM), as NP completions are generated much more often than Z ones. This indicates that both LMs strongly favor the NP analysis over the Z one when no disambiguating cues are given, with, in the vast majority of cases, virtually no syntactic uncertainty. The preference for the NP interpretation is in line both with preferences found in humans (Grodner et al., 2003) and other results on the behavior of LMs on NP/Z ambiguities (Futrell et al., 2019).

Despite this default preference for NP, when there is at least one cue that biases the prompt to the Z reading, $P(\mathrm{Z})$ spikes to 1 or close to it. This corresponds to the expected behavior on unambiguous prompts, where the Z interpretation is the only viable one. A qualitative inspection confirms that indeed most cases are correctly disambiguated (e.g.,

(1) In case the executive forgot the assistant a) *, the assistant was never fired and so on.* (NP) b) *explains the recommendations to this memo.* (Z)
(2) Because the train stopped the traffic was a) *much slower.* (Z) b) *suspended immediately.* (Z)
(3) Even though the girl phoned, the instructor a) *ignored her.* (S) b) *was too rude.* (S)

Table 5.5: Examples of completions generated by GPT2 for NP/Z prompts.

(2-3) in Table 5.5). The few completions labeled as NP are due to misclassifications or cases with unclear interpretation, analogously to those reported for NP/S.

However, around 8% and 25% of completions to Post-locus cue prompts of the LSTM and GPT2, respectively, exhibit behavior that we take to be indicative of an underlying confusion over the sentence structure, in spite of the Z interpretation being selected.[5] Concretely, the subordinate clause ends up ungrammatically incorporating two predicates, as exemplified by the following sentences (generated by GPT2; completion underlined):

(30) As the couple danced the tango began , the paparazzi swooned.

(31) Once the child played the piano was ours, it was somewhat expected.

An alternative grammatical interpretation of the sentence is possible if the comma joins two coordinate clauses, without a conjunction. However, the pervasiveness of this behavior with Post-locus cue prompts is suspicious: for comparison, in Pre&Post-locus cue, it only affects 0.01% of completions. Besides, all cases where the annotator used the label *unclear* were cases like this and were judged as ungrammatical. The occurrence of this phenomenon suggests that even when the Z reading is selected, the LMs may not fully adapt to its structure. The cause may be a lingering effect of the initially preferred NP analysis, or, more generally, difficulty in establishing the boundary between the subordinate and main clauses when it is not marked by a comma. Based on the aforementioned estimates, GPT2 seems to be affected more by this phenomenon.

All in all, the results indicate that with NP/Z ambiguities the LMs have a strong bias for the NP reading. With disambiguating cues for Z, the LMs pick the correct analysis, but, without a comma between the subordinate and main clauses, the LMs may still appear confused as to the structure of the sentence. That the LMs have difficulties in adapting to these temporary ambiguous sentences can be justified by the fact that typically a comma is indeed placed between a subordinate and main clause, even more so

[5] We identify these sentences through patterns in the dependency label: the post-locus cue is not recognized as the main verb, which instead appears after a comma, without an intervening conjunction (see Appendix C.3 for more details).

(1) Nobody knows if it's true that the university fines	
a) *are ever issued.* (Noun)	b) *people who don't study.* (Verb)
(2) Mrs. Baker is convinced that the school fears are	
a) *valid points.* (Noun)	b) *unfounded.* (Noun)
(3) Mary thinks that the pants suit me	
a) *better.* (Verb)	b) *really in a bad way.* (Verb)
(4) Despite last year's report, those city hopes	
a) *varied.* (Noun)	b) *to become wealthier.* (Verb)
(5) I know that this desert trains	
a) *people to work!* (Verb)	b) *are closed.* (Noun)

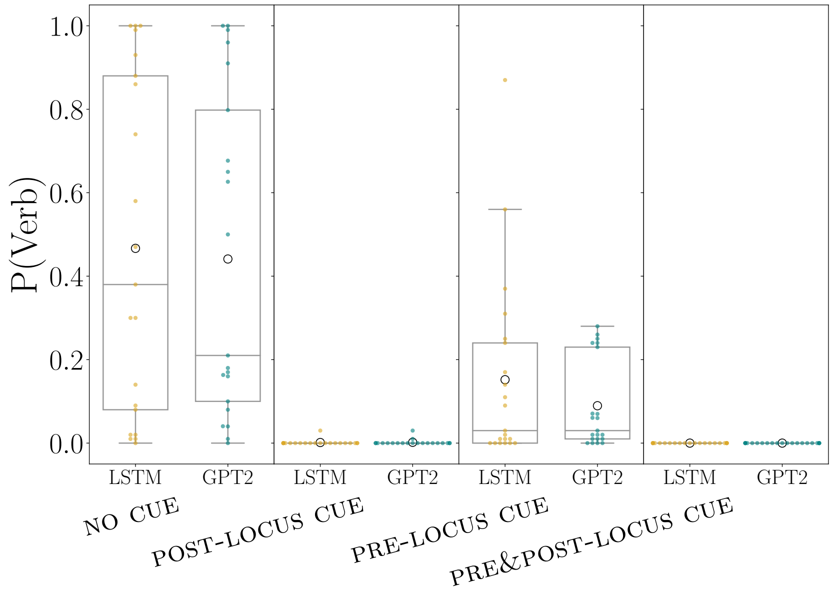
Table 5.6: Examples of completions generated by GPT2 for Noun/Verb prompts.

when this could lead to an ambiguity (in speech, these sentences can instead be disambiguated through prosody). The LM may then not observe sufficiently this construction in the training corpus to learn, on the one hand, that the Z interpretation is initially possible, and, on the other, how to fully recover when this eventually results to be the intended interpretation.

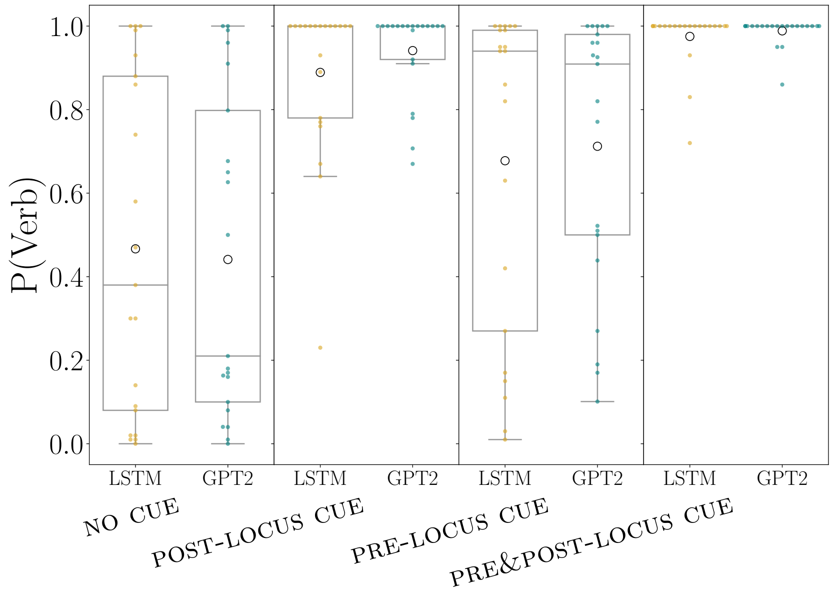
5.4.3 The Noun/Verb Ambiguity

Classification. To classify the generated sentences, we use the part-of-speech label predicted by the parser for the locus of ambiguity. When running the classification on the original sentence pairs, the classification sometimes fails, meaning that the tagger does not correctly interpret at least one of the sentences associated with that ambiguity (4 in total). To minimize noise, we discard the items where the parser failed, under the assumption that if that ambiguity is challenging to the parser already on these sentences, then errors are likely to spread also to the LM-generated sentences. This leaves us with 21 prompts for each prompt subtype. The agreement between the automatic labels and the annotator's ones is high (Cohen's $k = .83$). Differences occur due to tagging errors and sentences annotated as *unclear* by the linguist (5 cases).

Results. $P(\text{Verb})$ values for each prompt type are shown in Figure 5.3, while examples of completions can be found in Table 5.6. For ambiguous prompts – i.e., No cue – the dispersion of values is very high for both models, in particular even higher than what was found for NP/S No cue prompts. Though there is on average a preference for the Noun reading (mean $P(\text{Verb}) \approx .4$), the degree of syntactic uncertainty and the preferred interpretation vary across items. This indicates that the LMs' behavior on this temporary ambiguity is highly dependant on its instance. Since the prompts differ in the ambiguous



(a) Prompts with Noun interpretation



(b) Prompts with Verb interpretation

Figure 5.3: Distribution of $P(\text{Verb})$ for each Noun/Verb prompt type and LM. (a) and (b) report the results on prompts with Noun and Verb interpretation, respectively (the no cue condition is the same for both). Each dot represents the value for a prompt item, as estimated considering its completions sample; the circle indicates the mean across items.

word (the locus), an option is that the uncertainty depends on the lexical information the LM has acquired about that word. In this case, we could read $P(\text{Verb})$ as indicating the degree of dominance of the verb sense for the word, according to the learned lexicon of the LM. But it is also possible that the degree of bias towards an interpretation is also affected by the combination of the lexically ambiguous words and, for example, the previous one (e.g., “suit” vs. “pants suit”), with a more complex interaction between lexical and syntactic factors.

For Post-locus cue prompts, $P(\text{Verb})$ adapts to the disambiguating cues, approaching 0 and 1 for the Noun and Verb reading, respectively. In the latter case, we find some inter-item variation, due to some completions labeled as NP. A qualitative inspection shows that this tends to occur when a Noun reading is licensed despite the post-locus cue (e.g., “some metal rings loudly beat into our ears.”), or due to tagger errors. We generally do not find evidence of disambiguation issues on these prompts (e.g., (2-3) in Table 5.6).

By contrast, Pre-locus cue prompts, especially in the Verb subtype, pose more challenges to the LMs. $P(\text{Verb})$ values tend to follow the expected trends (decreasing for the Noun cases, and increasing for the Verb cases), but with much variation. In some Verb cases, we do not even find a preference for this reading (i.e., $P(\text{Verb}) < .5$). This behavior is not the expected one as the disambiguating cue constrains the correct interpretation. In particular, Pre-locus cue prompts are disambiguated by the number of the determiner. These results suggest that the LMs are not fully responsive to this cue, especially when pointing to a Verb reading. The LSTM shows a larger variation of values than GPT2, indicating greater disambiguation difficulty. A qualitative analysis confirms these observations: besides a portion of tagger errors, we find several completions that persist in the incorrect interpretation, thus violating number agreement (e.g., (4b) and (5b) in Table 5.6). This phenomenon can be explained by two scenarios. On the one hand, the LMs may have difficulty in capturing and tracking information on the number of the determiner (in particular the LSTM), and therefore it does not recognize the determiner as a disambiguating cue. Alternatively, the number of the determiner is actually kept track of, but the preference for an alternative interpretation eventually overrides its effect on disambiguation.

In general, we find that classification errors occur more often with the Noun/Verb ambiguity than with the previously analyzed ones. Lexical ambiguities are challenging for NLP systems, including when not causing temporary structural ambiguities (Elkahky et al., 2018). As errors introduce noise, our quantitative estimates can only be considered approximate. Yet, as mentioned earlier, the trends they point to are reliable as they are all confirmed by qualitative analyses of the data.

To summarize, on Noun/Verb temporary ambiguities the LMs’ uncertainty tends to strongly depend on the ambiguity instance. The LMs tend to be responsive to disambiguating cues, shifting their preferences towards the correct interpretation; however, some issues in the disambiguation arise if the only cue is the number of the determiner.

5.5 Follow-up Experiments

5.5.1 Effect of Decoding Strategy

In our previous experiments, we generated completions by sampling from the LM's output distribution. We compare this approach to other decoding strategies, focusing on NP/S No cue prompts.

Stochastic Decoding. Variants of stochastic decoding modify the LM output distribution before sampling words from it. Restricting or biasing the sampling process to high probability words can improve the quality of the generated text (Holtzman et al., 2019), while at the same time reducing the diversity of the text that can be generated from the prompt.

In **nucleus sampling**, the LM distribution is truncated to the set of top-probability words such that their cumulative probability is at least p – the nucleus size: words that do not fit in this set are assigned zero probability, and the distribution is recomputed.

$$\textstyle\sum_{v \in V^p} P(x_{i+1} = v | x_{1:i}) \geq p \tag{5.5}$$

$$\text{if } v \notin V^p, P(x_{i+1} = v | x_{1:i}) = 0 \tag{5.6}$$

$$P(X_{i+1}|x_{1:i})' = \text{softmax}\big(P(X_{i+1}|x_{1:i})\big) \tag{5.7}$$

The lower the value of p is set to the less words the probability mass is distributed over.

Another technique is that of re-shaping the distribution over words by dividing the output scores by a parameter t – **temperature** – before softmax is applied:

$$P(x_{i+1} = v | x_{1:i}) = \frac{\exp(o_v/t)}{\sum_{v' \in V} \exp(o_{v'}/t)} \tag{5.8}$$

If $t \in [0, 1)$, the distribution is skewed towards high probability words: the lower the temperature parameter, the more high-probability words become even more likely to be sampled, further flattening the values at the tail of the distribution.

We inspect how these decoding strategies affect the diversity of interpretations of completions to an ambiguous prompt. As p or t decreases, words in the tail of the distribution are less likely to be sampled. For different combination of values of p and t, we generate a set of completions from prompts and report the average $P(\text{S})$ (Table 5.7), following the procedure described in Section 5.3. Standard stochastic decoding, adopted in the previous experiments, corresponds to the setup where $p = 1$ and $t = 1$ (the output distribution of the LM is left unchanged). The values of $P(\text{S})$ decrease as the hyperparameters are modulated to focus on high-probability words (t or p decreases), indicating that a lower number of S completions get generated. This indicates that temperature and nucleus size can influence how ambiguities in an input get resolved in the text generated by a LM: Focusing on top-probability words increases the bias towards the preferred interpretation (NP, in the case of NP/S), which now more consistently manifest in the completions, thus limiting the "syntactic" diversity of the completions.

Maximization-based Decoding. In the previous experiments, we considered a sample of completions generated from a prompt to infer the LM's expectations over its analysis. Alternatively, one could look at the completion to which the LM assigns the highest probability as a way of establishing the overall preferred interpretation. Is the preference for an interpretation observed sampling multiple completions also reflected in the interpretation of the top-probability completion? To study this, we use **beam search** as maximization-based decoding strategy, returning the completion ranking highest in probability. Concretely, the beam search generation algorithm keeps track, at every timestep, of the beam of most likely sequences so far (we use beam size $= 16$). Since in this case we consider only one completion per prompt, $P(\mathrm{S})$ for beam search in Table 5.7 corresponds to the proportion of prompts whose top-completion has an S interpretation. The results show that most – though not all – prompts have an NP interpretation. This is in line with what was found in our experiments using stochastic decoding: in NP/S temporary ambiguities, the LMs have a general preference for the NP interpretation, but this does not apply uniformly to all instances as in some cases many S completions are also generated. It is however important to emphasize that, while maximization-based decoding can be seen as an alternative way of probing the preferred interpretation of a LM, it does not inform us about the degree to which this is expected. For this purpose, we require to explore the expectations of the LM more broadly, as we did when using stochastic decoding to generate a sample of completions.

5.5.2 Comparison to Surprisal Approaches

Previous work has probed the syntactic state of a LM in a temporarily ambiguous sentence by measuring the surprisal at the disambiguation point – underlined in (32) – (Futrell et al., 2019): If surprisal is higher than in the unambiguous sentence (33), we can infer that LM initially preferred the alternative, ultimately incorrect analysis. This is because it was not as surprised when instead a pre-locus disambiguating cue had already revealed the interpretation.

(32) The employees understood the contract *would* be changed very soon.

(33) The employees understood that the contract *would* be changed very soon.

Both this surprisal-based method and our generation-based one can be seen as ways of probing the syntactic expectations of a LM. Our method estimates the implicit prob-

	p	t	LSTM	GPT2
Pure sampling	1	1	.19	.27
Nucleus sampling	.9	1	.18	.24
	.75	1	.15	.23
	.6	1	.15	.22
With temperature	1	.9	.18	.24
	1	.75	.15	.24
	1	.6	.14	.23
Beam search - 16	-	-	.10	.25

Table 5.7: Average $P(\mathrm{S})$ for different decoding strategies on No Cue NP/S prompts (p: nucleus size; t: temperature).

abilities assigned to the candidate interpretations by the LM. Word surprisal scores do not measure these probabilities but can be seen as reflecting them: the more likely an interpretation of an input, the less surprisal difference at disambiguation should be found between the ambiguous and unambiguous versions of the input (e.g., between (32) and (33)).

Focusing on NP/S temporary ambiguities, we compare the two approaches to assess whether they are aligned in their estimates. On the one hand, we calculate the difference in word surprisal of the disambiguating word that follows the locus of ambiguity (i.e., the post-locus cue) between the ambiguous and unambiguous sentences of a pair (e.g., (32) and (33)). On the other, we calculate the $P(\mathrm{S})$ values on No cue prompts, following our generation-based method. For both LMs, there is a strong negative correlation between the two estimates (GPT2: Spearman's $\rho = -.70$; LSTM: $\rho = -.81$; both $p < .05$). As the probability of the S interpretation increase, the less difference in surprisal is found when the sentences is disambiguated as S – at the post-locus cue – in the temporarily ambiguous sentence and its unambiguous equivalent, respectively. This result corroborates that, even though surprisal estimates do not directly quantify expectations about syntactic interpretations, they do reflect them to a large extent, given how aligned they are to our estimated probabilities of interpretations. This moreover indicates that the insights observed with either method when probing the syntactic expectations of a LM are robust. In the next section, I further compare the two approaches, highlighting advantages of our generation-based approach.

5.6 Summary and Discussion

In this chapter, I have introduced a method to study the uncertainty of a LM over the interpretation of an input. Text generation was used as an analysis tool; in particular, to estimate a probability distribution over the candidate interpretations. We applied the method to study the resolution of temporary syntactic ambiguities, shedding light on the degree of preference of LMs for certain readings in both the absence and presence of disambiguating cues in the context. In the following, I first discuss the implications of our findings for the understanding of LMs' behavior, as well as avenues for future work. I then discuss potentialities and challenges of the methodology introduced.

5.6.1 Language Models and Syntactic Ambiguities

A large amount of previous research provided evidence that LMs take into account syntax to a large extent when processing an input (Linzen and Baroni, 2021). When the input is compatible with multiple parses, some works provided evidence that the LMs tend to have an initial preference for one of the interpretations, exhibiting a surprisal level at disambiguation mirroring garden-path effects in humans (Van Schijndel and Linzen, 2018; Futrell et al., 2019). However, these studies did not directly clarify the degree to which, in presence of ambiguities, the LM is uncertain among multiple interpretations. The results reported in this chapter contribute to this question by quantifying the extent to which each potential interpretation of an input is implicitly entertained by the model, both on ambiguous and unambiguous inputs.

Ambiguous Inputs (No cue). We found that when processing temporary syntactic ambiguities, LMs often exhibit uncertainty over the interpretation of the input; that is, they consider two candidate interpretations simultaneously viable (i.e., neither of them is assigned all – or nearly all – probability mass). This is displayed by the diversity of sentence structures in the completions generated by the LM. The presence of syntactic uncertainty was found for NP/S and Noun/Verb ambiguities. By contrast, on NP/Z ambiguities, a consistent and strong preference for the NP reading was found, indicating that the LM has learned to resolve these ambiguities via a general syntactic preference leading to essentially ignore the Z reading across cases. This can be justified by the infrequency of temporarily ambiguous NP/Z sentences which eventually result in a Z interpretation (without a comma marking the boundary between the subordinate and main clause; e.g., "Event though the band left the party it went on for another hour.').

On NP/S and Noun/verb ambiguities, syntactic uncertainty tends to be present but with much inter-item variation in terms of how likely an interpretation is considered. This dispersion of the values indicates that the way the LMs process these ambiguities, when no disambiguating cue is given, depends on the concrete instance, varying in the lexical items used to realize it (e.g., "The employees understood the contract..." vs. "The mechanic accepted the car..."; "the warehouse fires..." vs. "the pants suit..."). This suggests that the LMs take into account other factors other than just abstract syntactic preference when resolving these syntactic ambiguities, leading to a diverse behavior across cases.

Some evidence that this is the case at least for humans was provided by several studies. For instance, Garnsey et al. (1997) showed that in the processing of NP/S ambiguities there is an effect of the strength of the main verb bias (e.g., how often "understand" occurs with a direct object or sentential complement) and of the plausibility of the locus of ambiguity for an NP reading (e.g., "The employees understood the contract vs. the breakfast ..."). We used the NP/S items from Grodner et al. (2003), which all included main verbs biased for an NP reading based on subcategorization frequencies. We indeed found a general preference for the NP reading in the LM's behavior, but still with much variation in degree, with even some cases favoring an S reading.

The inter-item variation was even higher on Noun/Verb ambiguous prompts. The items from Frazier and Rayner (1987), used in our experiments, were not controlled for the dominance of the interpretations (e.g., how often "suit" is used as a noun vs. verb). In a post-hoc analysis, they collected judgments over the preferred interpretations for the items and found them to vary in this respect: roughly, half of the items favored the Noun reading, and half the Verb reading. Given this variation, it is not surprising that a LM's preferences also differ across cases. In particular, for humans, there is evidence that for lexical ambiguities the dominance of a reading influences the information that is initially activated during processing (Duffy et al., 1988; Rayner and Frazier, 1989). It is possible that the LM exhibit an analogous behavior with an initial bias to the most dominant interpretation of the lexically ambiguous word. Another factor that possibly affects the inter-item variation is plausibility or frequency of the locus of ambiguity and its preceding word as a noun-noun compound (e.g., "pants suit" vs. "warehouse fires").

More experiments could be carried out to better understand what factors drove the variation of syntactic uncertainty of the LMs across cases of NP/S and Noun/Verb ambiguities. First, one could test the behavior of the LM on more ambiguous prompts manipulating elements in them to test the effect of certain factors. For instance, for NP/S we can look at the effect of main verb bias (e.g., "understood/read the contract ...") and the plausibility of the locus of ambiguity for the NP reading (e.g., "understood the contract/the breakfast"). Moreover, one could assess whether the variation in the estimated degrees of preference of the LM matches structural frequencies in the training corpus (e.g., the main verb bias in NP/S ambiguity).

Finally, we experimented with different stochastic decoding strategies to assess how hyperparameters like nucleus size and temperature affect the way the ambiguities are resolved in the generated texts. We found that modulating these parameters can influence the interpretation of the input that is more likely to emerge during generation: if one interpretation was in the original distribution of the LM more likely, it becomes even more dominant when increasing the bias toward high-frequency words via the hyperparameters.

Unambiguous Inputs. We also investigated the behavior of the LMs when disambiguating cues are given as part of the input, before or after the locus of the ambiguity, respectively, or both. In general, the LMs tend to display the correct behavior. With some exceptions for Noun/Verb ambiguities, when looking at the distribution over interpretations, we observe the appropriate shifts in the probability with the correct reading becoming the most likely. This indicates that the LMs have a context-sensitive behavior and tend to correctly take into account the evidence in the input to properly disambiguate it. However, we also found evidence that disambiguation is not always carried out smoothly.

For post-locus cue NP/Z prompts, the lack of a comma between the subordinate and main clauses (pre-locus cue) led to some ungrammatical completions revealing an underlying confusion of the LMs over the sentence structure (e.g., "*As the couple danced the tango began, the paparazzi swooned"). This phenomenon can be explained by the LM downplaying the role of the post-locus cue (in the example above, "began"), thus not fully ruling out the NP reading. This issue affected the GPT2 model more than the LSTM, in spite of the former being bigger and trained on more data. An hypothesis is then that the difference might be due to the way the two models process the input, one through recurrent connections and the other through self-attention. This could be further investigated by, for instance, looking at the gate values and attention weights in the LSTM and GPT2, respectively. Nevertheless, the fact that recovering from a misinterpretation of an NP/Z ambiguity pose challenges is not completely surprising. It can be explained in terms of the higher complexity of readjusting the initial parse of the sentence (Grodner et al., 2003).[6] Issues in reanalysis were indeed also found for humans (Christianson et al., 2001). But also more simply, commas tend to be typically placed between a subordinate and main clause (alternatively, in speech the boundary is marked through prosody). It may therefore not be realistic to expect LMs to learn to cope with this type of construction.

The other disambiguation issues were found on Noun/Verb ambiguities, and in particular pre-locus prompts. In this case, the disambiguating cue was the determiner before the locus of ambiguity: for instance, in "this desert trains", "this" makes "trains" a verb due to the incompatibility between the singular determiner and the plural noun interpretation. From previous evaluations of LMs (Linzen et al. (2016) and subsequent

[6] In NP/S sentences, the initial verb phrase dominates the noun phrase that is the locus of ambiguity, independently of whether that is a direct object or part of a sentential complement –NP or S reading; by contrast, in NP/Z this is only the case if the locus of ambiguity is a direct object – NP reading. This can cause difficulty in readjusting the implicitly maintained sentence structure when encountering a cue to the Z reading only after the locus of the ambiguity.

work), we know that these models tend to account for number agreement in their output predictions: if they observed a singular subject, they tend to expect a verb with the same number. These works did not focus directly on the number of a determiner, but it is plausible that the ability of tracking number would generalize. Our results however show that LMs, especially the LSTM, do not always pick the correct interpretation when the number feature of the determiner acts as disambiguating cue (e.g, "*this desert trains are closed"). This aspect requires further investigation to disentangle whether the disambiguation is caused by a difficulty in tracking the number of the determiner, or by a strong preference for the alternative interpretation leading to downplay the role of number agreement for disambiguation.

Again, it should be noted that these ambiguities are particularly hard cases of lexical ambiguities, purposely designed to be ambivalent and to solely rely on number agreement for disambiguation. Due to this, these experiments may magnify the effect of disambiguation difficulties in LMs. Still, our results indicate that there is room for improvement in the LMs' responsiveness to contextual cues during processing.

5.6.2 Methodological Considerations

The methods used in this chapter and in Chapter 3 both aimed to analyze how LMs process ambiguities. The two methods however differ in that the former focused on the probabilistic output of the model, while the other on its internal representations. The behavioral evaluation we applied in this chapter did not require any supervision but probed the LM's ambiguity resolution by looking at its output behavior.

Most behavioral analyses of LMs focus on the probability assigned to a certain word in a sentence; for instance, to test subject-verb agreement, one looks at whether a higher probability is assigned to the correct than the incorrect form of a verb. (e.g., "the keys to the locker is/are"; Linzen et al. 2016); or to test the effect of implicit causality verbs, which pronoun – and thus referent – is more likely ("Bob congratulated Mary. He/She ..."; Davis and van Schijndel 2020). To extend this to the analysis of ambiguity resolution, one would have to identify a word after the locus of ambiguity which, if higher in probability than a contrast one, indicates that the correct interpretation is taken. Identifying such a contrast pair is however not easy, as there are countless ways in which an interpretation can be disambiguated.

(34) The employees understood the contract *well/./today...* (→ NP) vs. *ended/expired/was...* (→ S)

But the continuation to an input typically reveal its interpretation. Instead of focusing on the probabilities assigned to specific words, we proposed to explore the behavior of the model by sampling from its output distribution to generate completions to the input text. Our methodology extends the toolbox available to researchers when studying the resolution of ambiguities, which I hope will inspire and enable more research on this topic.

We applied this method to temporary syntactic ambiguities. Previously, these were studied looking at how expected – in terms of surprisal – the model was to encounter a certain word (or sequence of them) instantiating an interpretation (e.g., Futrell et al. 2019). Instead of comparing the surprisal score to that of a contrast word (as in (34)), the same estimate was computed for an unambiguous counterpart of the target sentence, analogously to experimental paradigms used in psycholinguistics. We compared our method to this approach in Section 5.5.2, and showed the two lead to highly correlated estimates. This indicates that the two methods are aligned in their findings and can be seen as alternative ways of probing the syntactic expectations of a LM. An advantage of the surprisal-based method is that, in comparison to our generation-based method, it is more straightforward to apply, as surprisal can be directly derived from the LM's output. However, our method, though involving a more complex procedure, has a number of advantages.

First, to study the ambiguity resolution of the LM, we directly infer the probability distribution over interpretations. Word surprisal may depend on this (and therefore correlate with our estimates; Sec. 5.5.2), but is not directly interpretable in this sense. Second, to study the behavior of a LM on an ambiguous input, the surprisal-based method requires comparing it to its unambiguous version. Our method does not require this and just has to be applied to the target input. This enabled us to inspect the behavior of the LM also on unambiguous inputs, leading to insights about the context-sensitivity of the model. Finally, letting the LM's expectations over a sentence manifest in concrete generated sentences can reveal or clarify aspects of the LM's expectations that would not be evident by looking at surprisal only (e.g., the disambiguation issues for NP/Z).

Notwithstanding, the methodology I presented in this chapter also has some challenges to be addressed by future research. Relying on an automatic classification of sentences was essential due to the large number of sentences that were generated for this study, but a computational parser may introduce errors, and thus noise, in the analysis. Though nowadays parsers are extremely accurate, it is precisely ambiguities – and even more so temporary ones – that can be challenging to them (Elkahky et al., 2018), even if they are explicitly trained to interpret expressions. We tried to limit the noise introduced by parser errors and to make sure that the automatic classification was sufficiently reliable to be used for the analyses. Even so, a more accurate method to classify the completions is desirable. Furthermore, with our current method, a completion was always assigned an interpretation (the parser always returned an output), but it would be useful if completions that are ungrammatical or have an unclear analysis could be flagged as such, to more easily spot disambiguation issues.

Though we only applied the methodology proposed to syntactic ambiguities, this could be extended to study further types of ambiguities. In fact, discourse continuation experiments have been used in psycholinguistics for other phenomena; for example, the way ambiguous pronouns are resolved (Stevenson et al., 1994; Kehler and Rohde, 2013):

(35) Bob congratulated John. *He* ...

  a) was very happy for him. (He = Bob)

  b) got the promotion. (He = John)

Analogously to our method, the completion is linked to an interpretation of the ambiguous element (“He”). We can therefore imagine using items of this kind as prompts to test the referential ambiguity resolution of LMs, or, analogously, other types of ambiguities, such as the lexical one in (36).

(36) I forgot about the *ball* ...

  a) though it is such an important event (ball = 💃)

  b) so we can’t play (ball = ⚽)

However, there are additional challenges in applying this method to semantico-pragmatic ambiguities. In local syntactic ambiguities, structural constraints rigidly enforce a certain interpretation. Semantic interpretations can instead be more flexible: they can be defeasible or vary across subjects. For instance, we can think of contexts where (36a) and (36b) lead to reversed senses (e.g., if one forgets a ball ⚽ for an important tournament, or if a band forgets they need to play at a ball 💃 event). It is possible that looking at longer sequences (beyond sentence boundaries) can help to further clarify the intended interpretation in these cases. As in our experiments, the analyses could be expedited by using NLP tools (e.g., coreference resolution or word sense disambiguation systems) for the automatic classification of completions.

Chapter 6

THE EFFECT OF CONTEXTUAL EXPECTATIONS ON REFERENCE PRODUCTION

This chapter investigates the interaction between reference production and referential ambiguities using deep learning models. By the resolution of a referential ambiguity, I refer to the process of interpreting a referring expression by linking it to the real-world entity it is intended to point to, when multiple options may be possible (Arnold, 1998; Nieuwland and Van Berkum, 2008). Referring expressions can vary in their degree of informativeness, and consequently their ambiguity potential, as exemplified by the following sentence (mentions highlighted with the same color corefer):

(37) Ann asked Lucy if she could passed her the bag.

The proper names "Ann" and "Lucy" introduce two female entities via labels that are specific to them and therefore unambiguous. However, the third-person pronouns "she" and "her" could refer to either of them based on gender agreement. The context can help us to correctly interpret them, out of an interaction between syntactic and semantico-pragmatic cues, and world knowledge (Hobbs, 1978; Grosz et al., 1995; Kehler and Rohde, 2013). The definite noun phrase "the bag" points to its referent through a description, making it a rather informative expression especially if the description cannot apply to more than one discourse entity. If that is not the case, to be even more informative (e.g., "the blue bag") might further ease the identification of the referent.

In this chapter, I address the hypothesis that the predictability of a referent in a discourse context affects the form chosen to refer to it. Concretely, that an ambiguous expression like "she" would be used where the context alone already makes a certain referent very likely; when this is not the case, a more informative, probably longer expression (e.g., "Mary Poppins", "the nanny") would be used instead (Tily and Piantadosi, 2009). We investigate this hypothesis on corpus data (covering ≈10K referring expressions) using a computational model that estimates how expected a referent is in a discourse context; that is, its predictability. For this purpose, I present a methodology taking advantage of NLP techniques, in particular state-of-the-art deep learning architectures for coreference resolution. This approach is related to that taken in Chapter 4 where I used deep LMs to derive representations of contextual expectations about word-level content. It, however, differs in that I here focus on expectations over what entity an expression refers to, and that these are computed by a model directly trained for this task (as opposed to a model trained on the generic language modeling objective).

6.1 Approach and Research Questions

The relation between predictability and form has been investigated by a long-standing research branch in linguistics. For instance, several works provided evidence for the hypothesis that more predictable information tends to be communicated with shorter, less informative words, or gets pronounced more quickly and prosodically attenuated (Ferrer i Cancho and Solé, 2003; Aylett and Turk, 2004; Levy and Jaeger, 2007; Piantadosi et al., 2011). An explanatory motive is that speakers aim to transmit information in an efficient way (Zipf, 1949): if some information can already be inferred from the context, they will reduce the cost of encoding their message. As more informative expressions pose typically more burden on production cost (being typically longer or less frequent), speakers will use these only when they provide essential information that would otherwise not be provided. In this way, they avoid redundancies between the informativeness of the context and that of the expression (Jaeger, 2010). The experiments presented in this chapter aim to test this hypothesis in the context of referring expressions.

Like other accounts of reference production, this study concerns the conditions that lead to choosing a certain form to refer to an entity (Davies and Arnold, 2019); in particular, contrasting third-person pronouns with the more informative alternatives (descriptions and names). Traditional accounts take referents to be associated with a certain discourse status determining the form which will be used to refer to them. Such discourse status has been connected to the *accessibility*, or *salience*, of an entity (a.o., Ariel, 1990; Gundel et al., 1993), and described as dependant on a series of discourse properties. Examples are the recency of the entity's last mention, competition (with other entities), syntactic prominence (whether it last occurred in subject position), or its topicality in the discourse. Within these approaches, a pronoun constitutes a pointer to a highly accessible entity. It is however not straightforward how to quantify accessibility.

Predictability – that is, how expected a referent is to be mentioned at a certain point in the discourse – was proposed as an alternative explanatory notion (Arnold and Zerkle, 2019). This property can also be referred to as **context informativeness**, in that if a referent is predictable from context only, it means that this provides much information. Referent predictability can be measured by eliciting which entity a subject would predict to be mentioned, only with access to the context, for instance through discourse continuation experiments or cloze tasks. For instance, a subject would be asked to continue an unfolding discourse like (38) revealing which entity they expect to be mentioned next (Stevenson et al., 1994); alternatively, the subject could be asked to point to the expected referent, using the antecedent in the discourse (Tily and Piantadosi, 2009).

(38) Barbara congratulated Lisa. ...

Current empirical evidence about the relation between predictability and the choice of a referring expression, however, paints a mixed picture. While some works using controlled items found that more pronouns were produced for more predictable referents (e.g., Arnold, 2001; Rosa and Arnold, 2017), other studies found that predictability was overridden by syntactic preferences (e.g. Fukumura and Van Gompel, 2010; Kehler and Rohde, 2013). Results using cloze tasks based on naturally occurring corpus texts also reported divergent results. Tily and Piantadosi (2009), analyzing newspaper texts, found that pronouns and proper names are preferred over full noun phrases when subjects were expecting the intended referent; but Modi et al. (2017), focusing instead on narrative stories, did not find this effect.

One of the challenges in getting a clear picture of the relation between predictability and form can be identified in the difficulty of scaling these analyses to larger and more diverse datasets. Some experiments tested the effect of specific factors like syntactic prominence, verb type, and rhetorical relations (e.g., Kehler and Rohde 2013; Fukumura and Van Gompel 2010). However, because of the items used, they do not address referent predictability in its full complexity, where many factors may be concurrently affecting expectations. Studies on spontaneously produced corpus data (e.g., Tily and Piantadosi 2009; Modi et al. 2017) tackle a more general notion of predictability without having to manipulate specific factors in their items. Still, the reliance on the collection of human judgments limits the scale at which these approaches can be applied.

We propose to expedite research focusing on corpus data through computational estimates of referent predictability, instead of human judgments in cloze tasks. We build on previous work in linguistics and cognitive science using computational methods to obtain predictability scores. This approach has been mostly employed in the study of language comprehension at the lexical and syntactic level; for instance, showing that predictability scores from LMs (e.g., surprisal) tend to correlate with measures of cognitive cost (Smith and Levy, 2013; Frank et al., 2013). LMs have also been used to study the trade-off between clarity and cost at the lexical level, reporting a cross-linguistic tendency for ambiguous words to appear in informative contexts (Pimentel et al., 2020a). Expectations at the referential level have received in comparison little attention, with the exception of a few works. Estimates of referent predictability were used by Orita et al. (2015) to explain referential choice as the result of an addressee-oriented strategy, in the context of the Rational Speech Act framework (Frank and Goodman, 2012). The measures of predictability were, however, only taking into account simple features like the frequency of an entity or how recently it was last mentioned. Modi et al. (2017) built upcoming referent prediction models considering shallow linguistic features and world knowledge about events and their typical participants. Through this methodology, they could make observations about the role of linguistic and common-sense knowledge, respectively, on referential expectations. However, their model required a substantial amount of annotations, both for training and evaluation, limiting its applicability for larger, and potentially out-of-domain, analyses.

To model referent predictability, we instead turn to a traditional task in NLP: **coreference resolution**. A coreference resolution model aims to group referring expression in a document based on the entity they refer to. For instance, in (37) the system would have to group together "Ann" and "her" as coreferring, by recognizing that the former is the anaphoric antecedent of the latter. The ordinary coreference resolution task cannot be directly used to model referent predictability, as it presupposes access to both the context of a referring expression and the expression itself. We adapt an existing deep learning architecture for coreference resolution (Joshi et al., 2020) to carry out a different task – *masked* coreference resolution – where the model can only use the information in the context to predict the referent of an expression. This can be seen as setting up a cloze task, analogously to Tily and Piantadosi (2009) and Modi et al. (2017), except that we now ask a computational model, instead of human subjects, to give us predictions. We train this model on an English corpus and then assess its abilities in both masked and ordinary coreference resolution. To further established the quality of our model as a proxy for referent predictability, we also assess how much the probabilistic output of our model matches human referential expectations, using data collected by Modi et al. (2017).

After this evaluation of our computational estimates of referent predictability, we study the relation between referent surprisal and the chosen referring expressions through a statistical analysis on the *test* portion of the English OntoNotes data. These data span a diversity of genres and allow us to analyze the phenomenon considering a vast amount of referring expressions. We look at mention form in terms of two aspects: the syntactic type of a referring expression (pronoun, proper name, full noun phrase), and its length (number of word tokens). This allows us to test whether shorter and less informative expressions, like pronouns, are used when a referent is more predictable, and vice-versa for longer and more informative expressions.

The chapter addresses the following research questions:

RQ1. How can we compute reliable estimates of referent predictability?

Analogously to the role of LMs in psycholinguistics (for instance, in the study of sentence processing; e.g., Levy 2008), having models that compute expectations at the referential level can be a helpful tool for linguists. Though it was shown that LMs take into account reference-level information at least to some degree (Sorodoc et al., 2020;

Davis and van Schijndel, 2020), they are not trained to identify referents and thus may not be optimal in their predictions. We opt for models that are directly trained to compute a probability distribution over entities. Concretely, we introduce a variant of the coreference resolution task to train a state-of-the-art deep learning architecture.

RQ2. Are less informative, more ambiguous expressions, like pronouns, used where the context provides more information about the intended referent?

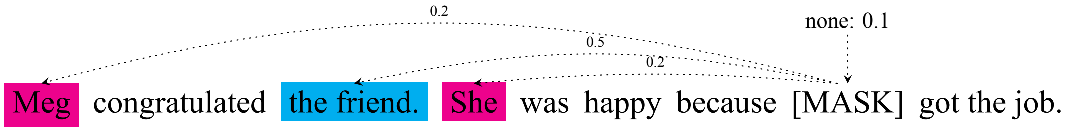
Expressions like third-person pronouns provide – out of context – little information about their intended referent (at best, number and gender). Yet, because they are short expressions, the speaker has an incentive to use them to reduce their production cost (Zipf, 1949). But if a speaker is collaborative with the addressee (Grice, 1975) and aims to reduce their challenges in interpretation, they should use pronouns only when the context complements their information, by making the referent predictable. The opposite could cause misinterpretations, which are the exception, rather than the rule, in communication. We thus expect to find pronouns in informative contexts.

RQ3. Are more informative expressions – descriptions or names – only used to compensate a limited informativeness of the context? Or do other factors affect reference production, on top of predictability?

To avoid redundancies, speakers may reserve the more informative expressions for less informative contexts, where the referent would not be otherwise identifiable (Jaeger, 2010; Tily and Piantadosi, 2009). But we could also find that speakers still sometimes opt for informative expressions (a name or a description, instead of a pronoun) in already informative contexts. For instance, a speaker may be overinformative if this does not come at a high production cost (e.g., "he" vs. "Tom"). We could then find that informative contexts are generally associated with short expressions, independently of the distinction across mention types (e.g., pronoun vs. name). We also consider discourse aspects that have been connected to the accessibility of an entity (Ariel, 1990), such as its recency of mention, syntactic prominence, etc.: While these factors may also play a role on referent predictability, they could have an even stronger effect on reference production. For instance, it was found that the tendency to use pronouns for subjects can override the role of predictability (Kehler and Rohde, 2013).

6.2 Masked Coreference Resolution: Methods

The objective of an ordinary coreference resolution system is to establish whether two referring expressions, or *mentions*, in a document *corefer*; that is, they refer to the same discourse entity. By doing so, they get to determine the coreference chains, or *clusters*, in the document, grouping mentions by their entity (Pradhan et al., 2012). Several deep



$P(E_{\text{[MASK]}} = \{\text{Meg, She}\}) = P(\text{antecedent}_{\text{[MASK]}} = \text{Meg}) + P(\text{antecedent}_{\text{[MASK]}} = \text{She}) = 0.4$
$P(E_{\text{[MASK]}} = \{\text{the friend}\}) = P(\text{antecedent}_{\text{[MASK]}} = \text{the friend}) = 0.5$
$P(E_{\text{[MASK]}} = \text{new}) = (\text{antecedent}_{\text{[MASK]}} = \text{none}) = 0.1$

Figure 6.1: An example of deriving referent probabilities from masked coreference resolution predictions.

learning approaches to coreference resolution have been proposed in the NLP literature. We focus on mention-ranking models, which carry out the task by outputting, for each mention, a probability distribution over its potential antecedents. These are the entities in the previous context, plus a “no antecedent” option (i.e., the current mention introduces a new entity). Once all mentions in a text are linked to their most likely antecedent, clusters of coreferring mentions are consequently determined.

Let us consider how referent predictability could be computed with mention-ranking model. The probability that a mention x refers to the entity e, $P(E_x = e)$, can be computed as the sum of the assigned antecedent probabilities of all mentions of e in the previous discourse (M_e; Figure 6.1):

$$P(E_x = e) = \sum_{i \in M_e} P(\text{antecedent}_x = i) \tag{6.1}$$

In a standard coreference resolution system, the model observes both the mention x and its context c_x to compute the predictions. It therefore calculates $P(E_x = e|x, c_x)$: a distribution conditioned on both the mention and its context. By contrast, referent predictability is about the degree to which a referent is expected based on context only (Tily and Piantadosi, 2009): that is, $P(E_x = e|c_x)$. How can we model this? We propose a variant of the coreference resolution task where referent predictability is estimated by accessing the context of a mention, but not the mention itself. To carry out this task, we propose a simple adaptation of an existing coreference resolution system. In the following, I introduce our model, and how its predictions can be computed and evaluated.

6.2.1 The Model

Model Architecture

We rely on the coreference resolution architecture by Joshi et al. (2020) (schematically represented in Figure 6.2) – henceforth **SpanBERT-coref**. The model builds on the ar-

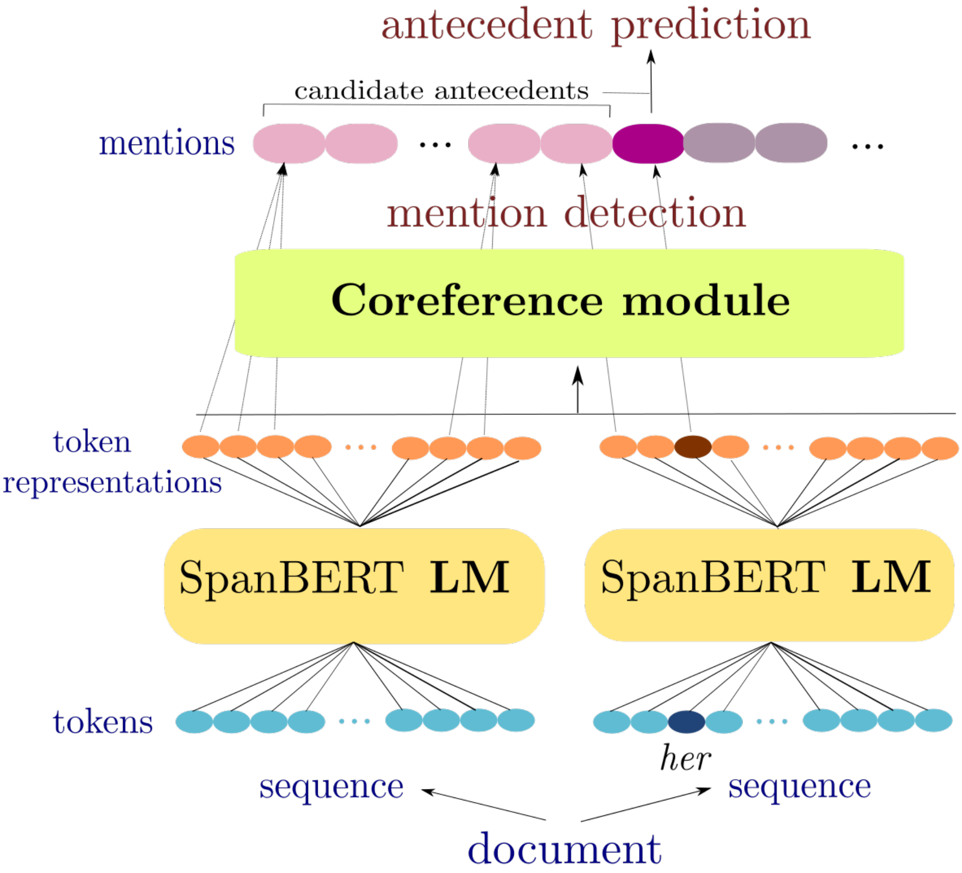


Figure 6.2: Schematic representation of the SpanBERT-coref architecture: token representations are obtained through the pre-trained SpanBERT LM, and then passed to the coreference module to detect mentions and identify their antecedents.

chitecture of Lee et al. (2018), which combines the benefits of transfer learning from LMs (Ruder, 2019) with an end-to-end approach to coreference resolution (Lee et al., 2017). The latter implies that both mention detection (i.e., identifying the referring expressions in the text) and coreference resolution are jointly carried out. While Lee et al. used ELMo (Peters et al., 2018a) as pre-trained LM, Joshi et al. (2019) adapted the architecture for use with BERT (Devlin et al., 2019), the transformer-based LM introduced in Chapter 3. SpanBERT-coref improved over these models by using instead a variant of BERT – SpanBERT – to enhance the representation of contiguous spans of tokens.

In both BERT and SpanBERT, a percentage of tokens is sampled for masking – substituted with the special token `[MASK]`; the model predicts these tokens based on their surrounding context (masked language modeling). In SpanBERT, however, contiguous sets of tokens –*spans* – are chosen for masking. Moreover, a Boundary Objective is added: to predict the words in a masked span at its start and end tokens.[1] This is to encourage that these maintain a representation of the span content. SpanBERT led to improvements over BERT when used for transfer learning to tasks involving spans of tokens, like coreference resolution. We, in particular, focus on the SpanBERT-coref model employing SpanBERT-base (12 hidden layers of 768 units, with 12 attention heads).

Besides the reliance on this concrete pre-trained LM, SpanBERT-coref follows the same architecture of Joshi et al. (2019); in particular, the *independent* version found to perform best. The model consists of two components: (1) the pre-trained LM, whose weights are fine-tuned during the training on coreference, and (2) a task-specific module on top of this, which is instead trained from scratch (Figure 6.2). Each document is split into sequences of a certain maximum length (always terminating at a sentence boundary). A span of text is represented by a fixed-length vector computed from the LM representations. In particular, each sequence is passed as input to the LM, and each token is represented by the last hidden state corresponding to its position. A span representation is derived as the concatenation of (1) the start token representation, (2) the end token representation, and (3) a weighted sum (attention) over the representations of all tokens in the span. Essentially, each span of tokens is assigned a contextualized representation putting together its lexical information – incoming from the word embeddings in the LM – and the context, which is, in this case, a window of surrounding tokens. With SpanBERT-base, a window of 384 tokens is adopted as found to perform best.

The span representations are used to compute the following scores: For each span, a mention score – how likely it is that the span constitutes the mention of some entity; s_m – and for each pair of spans, a compatibility score – how likely it is that the two expressions corefer;s_a. These scores are aggregated in a final score s for each pair of spans (Eq. 6.2), used to compute a probability distribution over the candidate antecedents of a mention (Eq. 6.3). The candidate antecedents are the identified mentions in the

[1] The Next Sentence objective, used in BERT, is dropped in SpanBERT.

previous discourse and “no antecedent”.

$$s(x, y) = s_m(x) + s_m(y) + s_a(x, y) \tag{6.2}$$

$$P(\text{antecedent}_x = y) = \frac{e^{s(x,y)}}{\sum_{i \in \text{candidate}_x} e^{s(x,i)}} \tag{6.3}$$

The original SpanBERT-coref model by Joshi et al. (2020) was trained on the English section of the OntoNotes v5.0 corpus (Weischedel et al., 2013); in particular, the portion with coreference annotations (1.6M word tokens). The data spans a diversity of genres: news, magazine articles, weblogs, religious texts, broadcast, and telephone conversations. Training followed the CoNLL-2012 Coreference Task (Pradhan et al., 2012), in both the data split (*train/dev/test*) and evaluation metrics (see Section 6.2.2). A set of 3.5K documents (1.3M tokens; 155K mentions) is used for training, and 343 and 348 documents are kept for evaluation during development and testing, respectively (160K and 170K tokens; 19K mentions each). For our experiments, we devise and train a variant of SpanBERT-coref. We retain the same architecture and training corpus, as well as the hyperparameters setup found to perform best for this model. As we explain next, to adapt the model to estimate referent predictability, we only manipulate the data that the model is fed during training.

Training on Masked Coreference Resolution

As explained earlier, to model referent predictability, we require a probability distribution over entities given the context – and not the mention – ($P(E_x|c_x)$), while a model like SpanBERT-coref gives us $P(E_x|x, c_x)$ due to access to the target mention.Our goal is thus to have SpanBERT-coref output predictions without accessing the mention. One way to achieve this is by not passing the mention information in the input, and ask the model to resolve the unknown mention using only its context, akin to the cloze task of Tily and Piantadosi (2009) and Modi et al. (2017). Models like BERT and SpanBERT, trained on a masked language modeling objective, already come equipped with a way of implementing a fill-in-the-gap task: When a token in the input is substituted by the `[MASK]` token it is predicted on the basis of its context, forcing the model to represent expectations about its content (as shown in Chapters 3 and 4). However, when the LM is used for coreference in SpanBERT-coref (or analogously in other tasks), the `[MASK]` token is not used again and all information from the input is available for coreference resolution

To estimate referent predictability with a system like SpanBERT-coref, we propose to *mask* the target mention in the input, that is, to substitute it with the `[MASK]` token. Only one `[MASK]` token is used to cover a mention, independently of its length, to fully omit all mention’s information (by contrast, in BERT or SpanBERT, a `[MASK]` is used to cover a single token).[2] The model will now compute $P(E_x|c_x)$ as only the context of the mention x is used for the prediction. We refer to this way of deploying a coreference model, displayed in Figure 6.1, as **masked coreference resolution**.[3] To mark the contrast, we refer to the standard task as **ordinary coreference resolution**. To implement either task, only the input to the model changes, while the architecture used to compute antecedent probabilities – and in turn referent probabilities (Eq. 6.1) – remains the same.

In principle SpanBERT-coref, though trained on ordinary coreference resolution, could be used on the masked version of the task in a zero-shot way (i.e., without changes to its training). This is because the SpanBERT LM component of SpanBERT-coref has encountered the `[MASK]` token during its pre-training. It should then be able to handle this setup at least to some extent by representing a masked mention in terms of information that is expected to fill its gap. However, since masked mentions were never passed to the model during training on coreference, we expect the predictions of SpanBERT-coref to be suboptimal proxies for referent predictability. First, the coreference component of the model, contrarily to the LM one, did not see masked mentions during training, and thus may not handle their representations well. Besides, because of a tendency of deep learning models to catastrophic forgetting (i.e., forget information about one task when later trained on another; McCloskey and Cohen 1989), it is not even guaranteed that the LM component can retain its ability to process masked tokens after training of coreference. Second, while during the masked language modeling training, `[MASK]` covered a single token, in masked coreference resolution it replaces an entire mention span, often longer than one token. As a consequence, SpanBERT-coref may be confused by expecting a single token in place of `[MASK]`. Finally, masked coreference resolution may require different or additional skills than those used for ordinary coreference resolution, such as paying more attention to certain contextual cues. This is empirically confirmed by the results reported in Section 6.3.

To build a more reliable proxy for referent predictability, we train a new instance of SpanBERT-coref which observes masked mentions during training and thus is optimized to handle this setup. I henceforth refer to the SpanBERT-coref model trained on ordinary coreference resolution as Coref model, and to our variant, trained also on masked coreference resolution, as Coref_M. To train Coref_M, at every epoch, a random sample of mentions in each document is masked; i.e., replaced by a single `[MASK]` token (re-sampling the mentions at every epoch). The percentage of mentions masked is a hyperparameter (we test values in the range 5%-40%): On the one hand, masking too many

[2] This is because types of referring expressions (e.g., pronouns vs. noun phrases) tend to differ in length, thus providing the model with information about mention form if we do not mask all mentions in the same way. As a sanity check, we also experimented with a variant of the system which masks a mention using a sequence of three `[MASK]` tokens, instead of one. This is to make sure that using just one token does not create any bias, to, for instance, be better on one-token mentions. The two ways of masking lead to analogous results, as reported in Appendix D.2.

[3] Further details about the masking procedure are reported in Appendix D.1.

mentions is expected to be detrimental, as too much information in the input is covered to properly solve the task. On the other, masking few mentions may not provide enough training signal to the model for learning to resolve them.

As the model is optimized to identify the correct antecedents of both masked and unmasked mentions, Coref$_{\mathrm{M}}$ should become able to output sensible antecedent predictions based on context only, or, when available, the context and the mention. However, resolving masked mentions is expected to be a harder task than resolving unmasked mentions, and, on some occasions, may even be impossible for humans. Because of its training, Coref$_{\mathrm{M}}$ should be better in masked coreference resolution than Coref, thus providing us with a better proxy of referent predictability. On ordinary coreference resolution, as both systems are trained to carry out this task, they may perform comparably.

For evaluation on *dev* data, to select the best model across training epochs and hyperparameters, we use antecedent accuracy on masked mentions (F1 score; see next section) as the criterion. We sample 10% of mentions in a document for masking (independently of the percentage used during training) and evaluate the predictions on masked mentions. We iterate the process 5 times (re-sampling masked mentions), finally taking the average of the evaluation scores. As I explain in more depth in the coming section, this evaluation is rather coarse-grained: it does not control for potential interference between masked mentions nor it considers masked predictions for all mentions. However, it gives us a good indication of the model's performances on masked mentions, while being quicker to compute than a more precise, but computationally more expensive, deployment of the model, which we reserve for evaluation and analyses on the *test* data.

6.2.2 Deployment and Evaluation

In the following, I describe the procedure used to deploy the system on a document, and to extract and evaluate predictions on masked and unmasked predictions, respectively. We use this procedure for our evaluation and analyses on the *test* portion of the OntoNotes data. Due to our focus, in our experiments, we mainly focus on a model's predictions on masked mentions. For a part of our evaluation we also however look at the quality of predictions on unmasked mentions.

Unmasked Mentions. By predictions on unmasked mentions, we refer to those obtained with access to both the mention and its context, corresponding to those used for ordinary coreference resolution. To extract these, we simply deploy the system of a target document as is, masking no mentions. To evaluate the coreference predictions, we follow the CoNLL-2012 shared task, and compute MUC, B^3, and CEAF metrics – as precision, recall, and F1 – focusing on their average as summarizing metric (Pradhan et al., 2012). These metrics focus on the quality of the predicted clusters of mentions (i.e., coreference chains) compared to the gold ones. The predicted clusters are obtained by linking each mention to the cluster of its most likely antecedent, or a new cluster if this is linked to no antecedent with most likelihood.

Masked Mentions. To extract predictions on unmasked mentions we can run the system on the document and extract all predictions at once, but we cannot do the same for extracting predictions on masked mentions. If we mask all the mentions in a document, the task of coreference resolution becomes virtually impossible, as too much information in the document is concealed. To model referent predictability, we aim at a setup as that in Figure 6.1: the target mention, whose referent is to be predicted, is masked, while the rest of its context, including the potential antecedents, are visible. However, to satisfy this scenario, we would have to pass a different input to the model for each mention of a document, changing the mention that is masked at each turn. We apply this method for the evaluation of the Modi et al. (2017) data (Section 6.4) where we compare the model's predictions to human judgments. For the larger-scale analyses on the OntoNotes *test* data (for evaluation and subsequent analyses of mention form), we instead employ an intermediate strategy, to achieve a trade-off between computational efficiency and interference between masked mentions.

Concretely, for each document, we identify a partition of the mentions, such that each subset can be simultaneously masked due to satisfying the following criteria: for each mention in each subset, none of its antecedents nor surrounding tokens (50 on both sides) is also masked. This ensures that all of the antecedents of a masked mention are accessible as well as the tokens in its local context, reducing the interference among simultaneously masked mentions.

Since only a subset of mentions is masked in a document, to evaluate predictions on these mentions the metrics used for ordinary coreference resolution are not suitable, as they conflate performances on masked and unmasked mentions by evaluating predictions at the cluster level. As we aim to evaluate predictions on unmasked mentions as proxies for referent predictability, what we care to evaluate is the quality of antecedent probabilities for a given target mention, as opposed to the quality of the cluster the mention is assigned to, which also depends on the model's output for other mentions. We then simply evaluate the system in terms of *antecedent prediction* quality, as precision, recall, and F1: a model's prediction for a mention is correct if it assigns the largest probability to a true antecedent (an antecedent belonging to the correct mention cluster), or to "no antecedent" if it is the first mention of an entity. We use F1 on antecedent prediction for masked mentions as the criterion for model selection during development. For comparison, we also evaluate antecedent prediction for unmasked mentions. A prediction relative to a mention is correct if it is linked to a true antecedent (belonging to the correct entity cluster), or to none if it is the first mention of an entity.

Mention Boundaries. As mentioned earlier, SpanBERT-coref is trained end-to-end to identify both mentions (i.e., their boundaries) and coreference links between them. The former challenge introduces additional uncertainty for the system when resolving the referent of a mention. Errors might occur due to a non-detected antecedent, or if predictions for a certain mention are not computed because this is not detected. As we aim to solely focus on the model's uncertainty over referents, we devise a strategy to bypass issues related to mention detection. We introduce a way of deploying the system where this directly uses the gold mentions and does not need to predict them. This setup is only used for deployment on *test* data and does not require any change to the way the model is trained. We pass the system the gold sentence boundaries and enforce that these are used as the only candidate spans for the output predictions. We then nullify the contribution of mention scores (i.e., how likely it is that a span is a mention) to antecedent predictions, by setting the mention scores to 0 for all mentions (i.e., $s(i, j) = s_a(i, j)$; Eq. 6.2). In this way, possibly incorrect expectations of the model about which spans constitute a mention cannot affect the output predictions. We compare the behavior of the system using predicted and gold mention boundaries, respectively: we expect the latter setup to lead to better modeling of referent predictability.

Context. As mentioned in Section 6.2, the SpanBERT-coref architecture builds on a bidirectional LM – SpanBERT – used to obtain span representations from an input linguistic sequence (Figure 6.2). For a target mention, only mentions in the previous discourse will be considered as candidate antecedents. However, the mention representation used to compute the prediction is derived considering the context that both precedes and follows the mention (for each in a degree that depends on the position of the mention in the sequence). For our experiments, we do not modify this aspect.

In this sense, our setup differs from the cloze tasks of Tily and Piantadosi (2009) and Modi et al. (2017), where participants were given only the context preceding a mention. In general, most studies on referent predictability or referential ambiguity resolution focus on the previous context of a mention. However, humans can also take the following context into account when interpreting referring expressions (Deemter, 1990; Song and Kaiser, 2020). We can think of cases as the following, where attending to the information coming after a mention (here, "she") can change its most plausible interpretation:

(39) Ann scolded her daughter, because

a) **she** was not behaving.

b) **she** was not happy with her behavior.

The role of the following context can be connected to a view where the resolution of referential ambiguities can be delayed or revised taking into account information that occurs after the mention. The speaker can take into account this in their choice of mention, as they know in advance – up to some limit – what they are about to say. The notion of referent predictability that we model is therefore not defined in a temporal sense as anticipation, but broadly in informational terms: how much information the context contributes to infer the intended referent of a mention if we set aside the information coming from the mention itself (Chapter 4 took a similar approach in modeling word-level expectations).

Besides this theoretical motivation, the choice of incorporating the following context is also practically an easier one at this stage, considering the state of the art in coreference resolution. Recent models tend to build on bidirectional architectures; sticking to this aspect allows us to obtain a good proxy for referent predictability by introducing minimal changes to these systems. To make Coref$_{\mathrm{M}}$ use a more controlled amount of context for each mention (its previous context, or, e.g., also the right context up to the end of the sentence) as opposed to a fixed-size window, can be done with these models. But we would have to pass a different input to the model depending on the mention of interest, which is computationally very inefficient, especially if done at training time.[4] Besides, there is no guarantee that the model could adapt well to this setup (e.g., with the sequence cropped right after the `[MASK]` token), since it was not trained in this way. We compute predictions based on the previous context only in the experiments on the Modi et al. (2017)'s data, reported in Section 6.4. However, in practice, we find this setup to lead to less human-like predictions than the left+context deployment of the model, despite more closely resembling how the data were collected. This is likely due to the mismatch between training and deployment of the model.[5] I discuss this aspect further in the section 6.4 and 6.6.

6.3 Masked Coreference Resolution: Results

Table 6.1 reports the results on OntoNotes *test* data for both masked and unmasked mentions. We compare the results of Coref (the existing coreference resolution model) and our variant Coref$_{\mathrm{M}}$. Based on evaluation on *dev* data, we select the Coref$_{\mathrm{M}}$model trained with 15% of mentions masked in each document.[6] For unmasked mentions, we provide results both for ordinary coreference resolution and antecedent prediction (Antecedent); for masked mentions, only the latter is applicable (see Section 6.2).

Results for both model-predicted and gold mention boundaries are reported. The

[4] The predictions relative to the mentions in a document cannot be computed by feeding the document once but passing a different version of the document for each mention cropped at its end boundaries.

[5] Experiments on antecedent prediction on OntoNotes led to similar conclusions.

[6] Models trained masking between 10%-35% of mentions achieved comparable antecedent accuracy on masked mentions on *dev* data. Among the best setups, we select for analysis the best model in terms of ordinary coreference resolution.

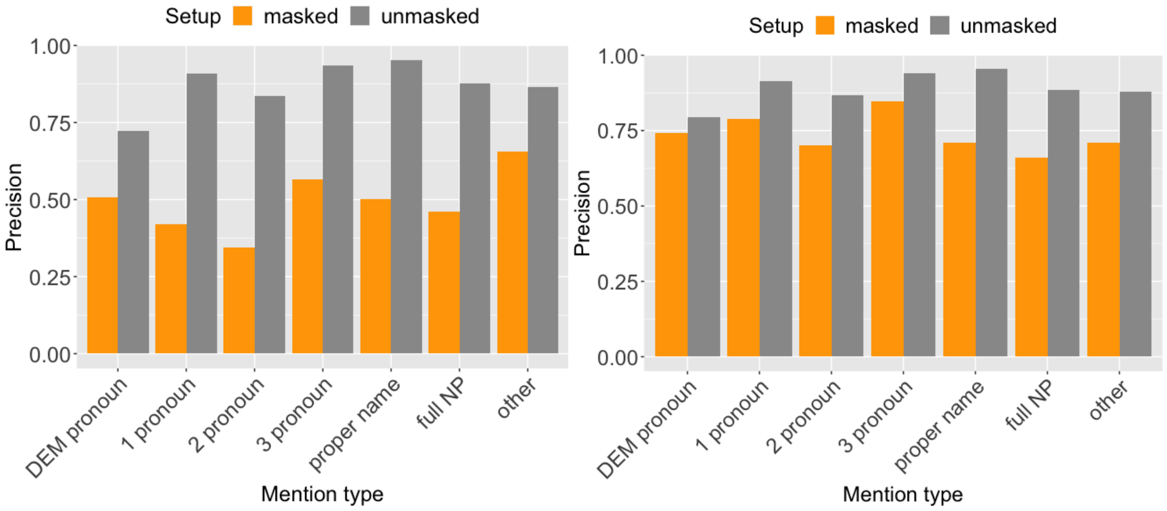
		Unmasked mentions						Masked mentions		
		Coreference			Antecedent			Antecedent		
	boundaries	P	R	F1	P	R	F1	P	R	F1
Coref	predicted	.78	.77	.77	.86	.82	.84	.42	.39	.4
	gold	.91	.85	.88		.90			.50	
Coref$_{\text{M}}$	predicted	.78	.76	.77	.86	.82	.84	.69	.69	.69
	gold	.91	.86	.88		.91			**.74**	

Table 6.1: Results on OntoNotes *test* data: coreference resolution (only with unmasked mentions) and antecedent prediction (both unmasked and masked mentions); P, R, F1 = precision, recall, F1 scores (when using gold mention boundaries on antecedent prediction, P = R = F1). BUC, M^3 and CEAF scores are reported in Appendix D.2.

latter method, across all scores, yields better results, in line with our expectations: removing the challenge of mention detection allows the model to output better antecedent predictions. Therefore, we use this setup in the analyses in Section 6.5 to obtain better proxies of referent predictability. Still, even with model-predicted mentions, the models' results are quite good, indicating that mention boundaries are typically identified correctly.

On unmasked mentions, the two models achieve essentially the same performances across metrics. This indicates that masking a subset of mentions during training, as we did for Coref$_{\text{M}}$, did not interfere with the learning of ordinary coreference resolution. On masked mentions, both models achieve worse results than on unmasked mentions. This is to be expected as the task is now harder: without accessing lexical information about the mention, the models can only use the context for the prediction, and this may not always constitute sufficient information. Nevertheless, the two models guess correctly on a non-trivial amount of cases. Even Coref, which was only trained on unmasked mentions, makes correct guesses on half of the mentions when using gold mention boundaries. A random baseline would only correctly resolve .08%, while selecting always the immediately previous mention or always "no antecedent" would achieve .23 and .26%, respectively. This supports the intuition that the knowledge transferred from masked language modeling (predicting a word covered by `[MASK]`) provides a good starting point to approach masked coreference resolution, as the `[MASK]` token is not new to the architecture

Still, as we anticipated, we achieve better performances with a model that resolved masked mentions also during training: Coref$_{\text{M}}$ improves substantially over the results of Coref on masked mentions. As mentioned earlier, introducing masks during training can help the system to adapt to this setup, including potentially enhancing the context-sensitivity it requires. In 74% cases, a correct antecedent prediction is computed based



(a) Coref

(b) Coref$_{\mathrm{M}}$

Figure 6.3: Antecedent precision for Coref and Coref$_{\mathrm{M}}$, respectively, across more fine-grained mention types, for masked and unmasked mentions.

on context alone (gold mention boundaries): in a large number of situations, the context is on its own sufficiently informative to identify the referent of an expression.

Results by mention type. In Figure 6.3 I report the antecedent prediction results of the Coref and Coref$_{\mathrm{M}}$ models across mention types. I focus on the results with gold mention boundaries, as this is the setup employed in subsequent experiments. Using predicted mentions, the magnitude of scores is lower but the relative differences across mention types are the same (results reported in Appendix D.2). We distinguish mentions that are proper names, full noun phrases (NPs), and different types of pronouns. Mentions that are proper names point to an entity with a label that is typically unique for that referent, therefore being very informative. NPs may vary in their informativeness by describing with more or less detail the entity that they refer to (e.g., “the student” vs. “the student that is sitting by the window”). The heterogeneity of pronouns may lead to differences in the behavior of the model and are therefore subclassified. First- and second-person pronouns are indexicals and therefore comparatively more rigid than other pronouns (they typically refer to the speaker and addressee in dialogue or reported speech). Other pronouns, like third-person ones or demonstratives (DEM), are instead inherently ambiguous and appeal to information in the context to be resolved. In describing the results, I mainly focus on third-person pronouns due to these being the subcategory of pronouns analyzed in Section 6.5, in line with previous work focusing on third-person referents (other than the interlocutors).

context	mention
(1) Judy Miller is protecting another source [...] Let me get a response from Lucy Dalglish. I think it's very obvious from what Judy wrote today **[MASK]** is protecting somebody else. ✓	she
(2) This child [...] felt particularly lonely and especially wanted his father to come back. He said that he was sick one time. **[MASK]** worked in Guyuan ×	his father
(3) One high-profile provision [...] was the proposal by Chairman Lloyd Bentsen of the Senate Finance Committee to expand the deduction for individual retirement accounts. **[MASK]** said he hopes the Senate will consider that measure soon ✓	Mr. Bentsen
(4) Sharon Osbourne, Ozzy's long-time manager, wife and best friend, announced to the world that she'd been diagnosed with colon cancer. Every fiber of Ozzy was shaken. **[MASK]** had to be sedated for a while. ×	he

Table 6.2: Examples of correct and incorrect predictions by Coref$_\text{M}$ (with gold mention boundaries) on masked mentions; model's prediction underlined, correct antecedent with dotted line.

Their behavior of Coref and Coref$_\text{M}$ is aligned on unmasked mentions: the two reach comparable performances also when splitting results by mention type. On masked mentions, Coref$_\text{M}$'s results are better for all mention types, indicating that seeing masks during training benefits the model across classes of referring expressions. In spite of magnitude differences for masked mentions, in general, the trends of which mention types pose more challenges to the two models are analogous: When looking at unmasked mentions, proper names are the easiest to resolve, while demonstratives are the hardest. By contrast, for masked mentions, third-person pronouns are the easiest, followed by proper names and full NPs.

The fact that the trends across mention types differ for masked and unmasked mentions suggests that, depending on the mention type, accessing the mention may be more or less crucial to identify its referent. For instance, Coref$_\text{M}$'s performances on ambiguous third-person pronouns are not so different when accessing the mention or not (unmasked vs. masked mentions). This can be explained by pronouns occurring in informative contexts and the limited information that the pronoun itself encodes. By contrast, for expressions like full NPs and proper names, there is a substantial difference in accuracy when the mention is revealed or not, indicating that in those cases the mention tends to often provide crucial complementary information. Table 6.2 shows examples of predictions on masked mentions with different mention types.

	accuracy	relative accuracy	JSD
Coref_M left only	.54	.50	.46
Coref_M left + right	**.74**	**.64**	**.39**
Modi et al. (2017) ling.	.59	.38	.57
Modi et al. (2017) ling. & world	.62	.53	.50

Table 6.3: Evaluation of Coref_M in terms of accuracy (with respect to the true referent), relative accuracy with respect to human top guess, average Jensen-Shannon divergence (JSD) of the probability distributions over referents derived from human predictions and from the model (the smaller the better).

6.4 Predicting Human Referent Expectations

Before using Coref_M as a proxy for referent predictability, we evaluate whether its predictions are human-like. In the previous evaluation, we assessed how often the model assigned the highest probability to a true coreferring expression (accuracy). However, humans are not expected to always be able to guess the true referent based on context. We thus evaluate Coref_M's expectations over referents in terms of the extent to which they approximate those that a human would have, including when they are not able to guess the true referent. To evaluate whether Coref_M is good at this, we compare its estimates of referent predictability with those derived from experiments with human subjects.

For this analysis, we rely on the human data collected by Modi et al. (2017) in a cloze task setup. These data come from the InScript corpus (Modi et al., 2016), a collection of narrative stories around different common scenarios (e.g., going grocery shopping, taking a bath). Due to this focus, these data trigger expectations that are strongly connected to world knowledge about events and their participants. Subjects were asked to guess the antecedent of an undisclosed mention while seeing only its left context (182 stories, $\sim$3K mentions with 20 guesses each). For example:

> ... I mixed the ingredients in a bowl, first sifting the dry ingredients together, and then beating in the butter, eggs, and sugar. I baked the cake at 350 degrees for 25 minutes, then took it out of the oven, let it cool, and frosted ___

From the antecedent guesses, one can derive a probability distribution over referents.

To elicit human judgments of referent predictability, Modi et al. (2017) used the syntactic head of mentions as opposed to the complete mention boundaries (e.g., "ingredients" in "the ingredients"). To make the task closer to the way the model was trained (using full mention spans as pointers to mentions), we identify the mention boundaries associated with each head through an automatic method, using detected *noun chunks* though the spaCy library and a set of heuristics (e.g., for undetected noun phrases, the preceding – if any – determiner is incorporated). This leads to an estimated 91% accuracy, based on a manual check of a sample of 200 mentions. We set the identified mention boundaries as gold mention boundaries, as previously explained. While these are used as candidate antecedents of a target mention, we set up the cloze task in the way that Modi et al.'s did, namely masking only the head of the target mention.

To obtain predictions given a target mention, we only mask this one in the document. We deploy Coref_{M} in two setups: 1) using just the previous context of the target mention (left), the setup used to elicit the human judgments, and 2) using both the left and right context of the mention (left + right), the setup used to train the model. To evaluate the model's estimates, we follow Modi et al.'s approach and compute Jensen-Shannon divergence (JSD) – a measure of dissimilarity between probability distributions – to measure the difference between the predicted and the human distributions over referents. The lower the divergence, the better the distribution estimated by the model matches that of humans. We also report the accuracy of predictions with respect to the true referent, and the relative accuracy with respect to the human top guesses.

We compare our results to those reported by Modi et al. (2017) for their computational models. These were trained to predict upcoming discourse referents based on 1) linguistic knowledge alone (ling.), and 2) both linguistic and world knowledge (ling & world). The models learned from the training portion of the InScript corpus, leveraging, among others, script annotation such as participant and event types. For linguistic knowledge, they use shallow features, like whether a referent was mentioned as the last subject, or more complex ones, like how much the referent fits the selectional preferences of the given syntactic position. For world knowledge, they focus on both whether a referent (associated with a participant type) fits the current position, and on prototypical sequences of events.

Table 6.3 reports the results. Our JSD results improve (lower scores) over those reported by Modi et al. (2017) for their models, indicating that we obtain a better proxy for human referential expectations. Modi et al.'s systems were built incorporating much information (syntactic and semantic) and trained in-domain. By contrast, our model only leveraged the coreference annotations in a larger but more diverse training corpus (OntoNotes). Our model gives better approximates of human expectations, while it provides the flexibility of adapting to texts of different genres, without requiring fine-grained annotations.

The comparison between the left and left + right deployment of Coref_{M} offers interesting methodological insights, albeit negative. Coref_{M} can be deployed with varying amounts of context. However, as emphasized earlier, the model was trained with access to a bidirectional context window, and may then not work well when the text is immediately cropped after the masked mention (left). Indeed, Coref_{M}'s predictions are more aligned to those of humans when accessing both sides of the context than with only the left context, even though the second setup more closely resembled that used for the human data collection. Information in the following context of the mention could not influence the human judgments, as it was not available. Therefore, if the left-only deployment of Coref_{M} worked as it should, we should not expect the left+right one to do better at approximating the human-derived distribution over referents. This is because both setups give access to the relevant information that humans had for their prediction – the left context: if anything, adding more information – the right context – could even lead to a deviation from human judgments because of the reduced uncertainty. Therefore, we take the results to indicate that Coref_{M} works suboptimally when deployed in a setup different from that used during its training, with the system possibly having difficulties in identifying the context cues available. For the current experiments, we focus on the model leading to the best approximation of human expectations so far, namely Coref_{M} in its left + right deployment.

6.5 Predictability and Mention Form

In the previous sections, we evaluated the ability of Coref_{M} to form referential expectations. Given that the evaluation provided satisfactory results, we here use the estimates of referent predictability from our computational model as proxies for the true expectations, to investigate the relation between predictability and mention form. Our analyses are analogous in methods to those of Tily and Piantadosi (2009) and Modi et al. (2017), except that we use computational estimates of referent predictability instead of human judgments.

Following previous work, we measure predictability with the information-theoretic notion of **surprisal**: the more predictable an outcome is, the lower the surprisal when it occurs. Given a masked mention x with its true referent e_{true}, surprisal is computed from the Coref_{M} probability distribution over entities E_x (Eq. 6.1), given the context c_x:

$$\text{surprisal}(x) := -\log_2 P(E_x = e_{\text{true}} \mid c_x)$$

Surprisal ranges from 0 – when there is no uncertainty and the correct referent is assigned probability 1 – to positive infinity if the true referent was assigned probability 0. We compute surprisal deploying Coref_{M} with gold mention boundaries, which, besides simplifying the prediction of the referent, ensures that the probability distribution over antecedents output by the model considers all and only the true set of candidate mentions.

Using estimates of referent surprisal, we address the questions of whether ambiguous expressions like pronouns are used where the referent is more predictable (RQ2), and whether informative expressions are used exclusively in cases where the context does not lead to strong expectations about the intended referent (RQ3; in other words, if redundancies are avoided). To classify mentions by their informativeness, we use a mention's syntactic type (third-person pronouns, full NPs, and proper names) and length (number of word tokens). We then analyze referent surprisal as a predictor of each of the two.

For these experiments, we use the OntoNotes *test* data, extracting for each (masked) mention the referent predictability estimates, as well as a set of relevant features, some of which leverage syntactic annotations in the corpus. First and foremost, to allow for the analyses, the type and length of a mention; then a series of shallow linguistic features:

- **distance**: number of sentences between target mention and its closest antecedent (inverse of recency);
- **frequency**: number of mentions of the target mention's referent so far;
- **antecedent = previous subject**: whether the closest antecedent of the target mention is the subject of the previous clause;
- **mention = subject**: whether the target mention acts as a subject;
- **antecedent type**: whether the closest antecedent of the target mention is a pronoun, proper name, or full NP.

The features are used to investigate whether surprisal can explain the choice of mention type on top of discourse factors previously connected to reference production (Ariel, 1990; Arnold, 1998). Coref$_{\mathrm{M}}$ learned a notion of referent predictability abstracting over data, thus autonomously identifying the information that seemed relevant to make good predictions. It is plausible that it learned to attend to some of these linguistic features, as they can make an entity more expected (e.g., an entity mentioned last is expected to be mentioned again; Arnold and Zerkle 2019). However, it may also have learned to leverage other cues, not captured by the features; for instance, semantico-pragmatic cues like the influence of thematic roles or rhetorical relations, which were shown to affect expectations (Stevenson et al., 1994; Arnold, 2001; Kehler et al., 2008).

6.5.1 Predicting Mention Type

We employ a three-way distinction of the syntactic type of a referring expression, following previous work: namely, third-person pronouns (henceforth referred to as simply *pronouns*), proper names, and full NPs. We thus exclude all other mentions, and also those without an antecedent (i.e., first mention of an entity). The relevant data amounts to 9758 datapoints ($\approx$4K pronouns, $\approx$2K proper names, and $\approx$3K full NPs). In Figure 6.4, the distribution of referent surprisal for each mention type is displayed: Full NPs tend to be associated with lower referent predictability (higher surprisal) than names and pronouns. Much in-type variation is observed, but pronouns more consistently stay in

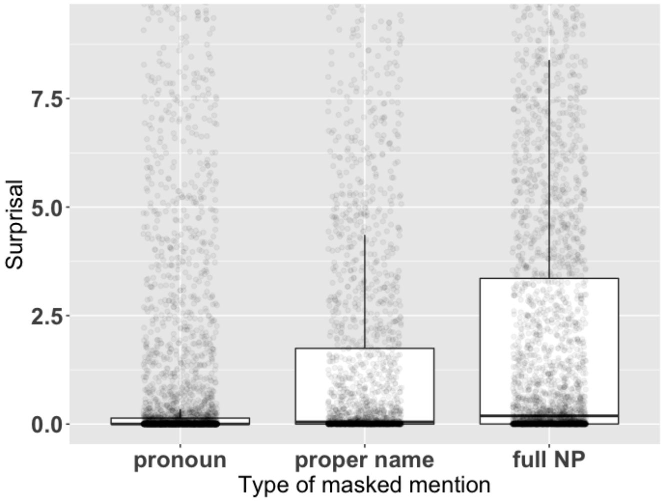


Figure 6.4: Surprisal and mention type. The limits on the y axis were scaled to the 95th percentile of the data to visualize the variability better.

the lower end of the distribution, meaning that they tend to be associated with lower surprisal.

We use multinomial logistic regression to quantify the effect of predictability on mention type, using as the dependent variable mention type with pronoun as the base level, and surprisal as independent variable.[7] Continuous predictors were standardized, allowing for comparison of coefficients. The results of this regression are given in the top left segment of Table 6.4. On the basis of the coefficients, we can see that higher referent surprisal increases the probability of a proper name ($\beta = .31$) and even more so of full NP ($\beta = .47$). This indicates that pronouns are used in contexts making the intended referent more predictable than proper names and full NPs.

Following Tily and Piantadosi (2009) and Modi et al. (2017), we test whether referent surprisal has any effect when we add as further predictors the aforementioned linguistic features, taken to play a role in mention choice. This allows us to test whether referent surprisal plays a role beyond what is captured by those features. We fit a new multinomial regression model, this time including also the linguistic features as independent variables. Through the Likelihood Ratio chi-squared test, we verify that incorporating each predictor improves goodness-of-fit (with standard .05 alpha level; see Appendix D.3 for more detailed results). Results are shown in the lower part of Table 6.4. Surprisal led to improved goodness-of-fit ($p_{\chi^2} << 0.001$), indicating that it still contributes information that is relevant to the prediction of the mention type and is not captured by the linguistic features alone. However, now that the linguistic features are added, surprisal does not distinguish anymore between pronouns and proper names,

[7] We use the *multinom* procedure from the library *nnet* (Venables and Ripley, 2002).

but is only predictive of the distinction between pronouns and full NPs (see significance values for surprisal in Table 6.4). This result was also found by Tily and Piantadosi (2009).

As for the linguistic features, pronouns are favored over proper names and full NPs when the referent has been mentioned recently, in line with the idea that pronominalization is sensitive to the local salience of an entity. Furthermore, a pronoun is more likely if the entity it refers to was last mentioned also with a pronoun. Proper names exhibit an analogous effect, with a tendency to being reused to refer to an entity. The results are also in line with the subject preference for pronouns (Arnold, 1998), as pronominalization is more likely when the referent's previous mention appeared in the subject position.

As mentioned earlier, we can take the information captured by the linguistic features to be connected to predictability. However, Coref_{M} could have acquired sensitivity also to other types of information, such as semantico-pragmatic aspects (e.g., verb semantics, rhetorical relations, world knowledge about events). Under this view, the results indicate that the overlap of the estimates of referent surprisal with the linguistic cues is only partial (adding one or the other leads to improvement in goodness-of-fit), and the choice of mention type is best explained as a combination of these factors. This may be because the predictability estimates fail to adequately account for the role of those linguistic factors as they should, or because those factors sometimes override the role of predictability when choosing mention form. I discuss further these results in Section 6.6.

Overall, the results of this analysis are in line with the findings from Tily and Piantadosi (2009). They also used data from OntoNotes but focused on a comparably small dataset of just news text. Thanks to the use of a computational model to estimate predictability, we generalized their findings on a larger scale, considering the larger OntoNotes *test* data, spanning a diversity of genres.

6.5.2 Predicting Mention Length

We codify mention length in terms of the number of word tokens in the referring expression.[8] This is linked to informativeness, under the view that using more words to refer to an entity leads to providing more information for its identification, though possibly with some degree of simplification.[9] But mention length can also be connected to cost: to produce a longer mention entails more production cost for the speaker, which can be justified by the intention of providing more information. While pronouns are

[8] Analogous results were found when codifying mention length as the number of characters.

[9] The degree of informativeness of an expression may not only depend on the number of word tokens, but also on the degree of specificity of the concept this encodes. For instance, "the satchel" and "the leather satchel" are more informative than "the bag" or "the leather bag", respectively. Moreover, the degree of informativeness of an expression varies contextually: "the bag" is more or less informative depending on the number of bags mentioned in the discourse.

		Predicting mention type							
		Proper name				Full NP			
		β	s.e.	z	p	β	s.e.	z	p
Intercept		-.63	.03	-23.8	-	-.26	.02	-10.9	-
surprisal		.31	.03	9.6	*	.47	.03	16.4	*
Intercept		-.24	.07	-3.6	-	.04	.07	.6	-
distance		3.13	.12	25.4	*	3.10	.12	25.2	*
frequency		.09	.03	3.1	*	-.13	.03	-3.8	*
antecedent =	previous subject	-1.31	.09	-13.9	*	-1.10	.08	-13.7	*
mention =	subject	.07	.07	1.0	.3	-0.50	.06	-7.7	*
antecedent type =	proper name	1.78	.08	22.8	*	.41	.09	4.6	*
	full NP	-.17	.08	-2.2	*	1.18	.06	18.1	*
surprisal		.05	.04	1.5	.1	.23	.03	7.8	*

Table 6.4: Two multinomial logit models predicting mention type (baseline level: "pronoun"), based on 1) surprisal alone and 2) shallow linguistic features + surprisal. *: significant predictors with $p < .05$. All predictors reported improved goodness-of-fit to the data. Complete tables with Likelihood-ratio Chi-squared test and F-test are reported in Appendix D.3.

		Predicting mention length			
		β	s.e.	t	p
Intercept		1.87	.02	80.8	-
surprisal		.25	.02	10.7	*
Intercept		1.81	.05	40.1	-
distance		.17	.02	7.1	*
frequency		-.13	.02	-5.4	*
antecedent =	previous subject	-.51	.06	-8.5	*
mention =	subject	.04	.05	.8	.4
antecedent type =	proper name	-.21	.06	-3.2	*
	full NP	.42	.06	7.5	*
surprisal		.17	.02	7.4	*

Table 6.5: Two linear regression models predicting mention length (number of tokens) of the masked mention, based on 1) surprisal alone and 2) shallow linguistic features + surprisal. *: significant predictors with $p < .05$. All predictors in models improved goodness-of-fit to the data except "mention = subject" in the full linear regression model. Complete tables with Likelihood-ratio Chi-squared test and F-test are reported in Appendix D.3.

always one-token mentions (e.g., “she”, “themselves”), names and full NPs can vary (e.g., “Mary Poppins” vs. “Mary”; “the nanny” vs. “the nanny hired by Mr. Banks”), giving us a more graded measure than the coarse-grained distinctions across mention types. On average, proper names are shorter than full NPs (mean number of tokens: 1.67 vs. 3.16).

Following the same approach employed in the analysis of mention type, we fit regression models with mention length as the dependent variable, (or number of characters, in the Appendix), and, and surprisal with and without shallow linguistic features, respectively, as independent variables. In this case, we use linear regression models due to length being a numerical variable. Table 6.5 reports the results. We find essentially the same trends for mention length which we found for mention type: More surprising entities tend to be referred to with longer expressions. When we add linguistic features as predictors, surprisal still contributes information on top of them. Among the features, the grammatical function and type of the antecedent are the strongest predictors. Figure 6.5 visualizes this trend between surprisal and predicted mention length.

The fact that pronouns are always one-token mentions raises the question of whether perhaps the linear regression is only capturing the distinction between pronoun and the rest of mentions. To address this, we fit a regression model with the same method but only considering the non-pronominal mentions: if the relation between length and referent surprisal still holds, this means that it applies generally to referring expressions, including names and full NPs. We find that this is the case, as the same results are found also when pronouns are excluded (see tables 4 and 6 in Appendix D.3): shorter expressions are associated with more informative contexts, independently of mention type.

Overall, these results suggest that speakers take into account the predictability of the entity they want to refer to when planning their referring expressions. Longer, more informative expressions are used to compensate a limited context informativeness. This can be explained by speakers going out of their way to produce a more costly expression when it would be otherwise difficult to understand the utterance. On the contrary, shorter expressions are favored in informative contexts. In the analysis of mention type we found that referent predictability did not capture the distinction between pronouns and names, when considering also the role of linguistic features (Table 6.5). The results on mention length can help us to posit an explanation for this result, as names tend to be shorter expressions than full NPs. It is possible that in informative contexts, a speaker may still opt for a more explicit expression than a pronoun when the redundancy this introduces does not cause much production cost: even though the referent can be inferred from the context, the speaker does not use a pronoun but, e.g., a name.

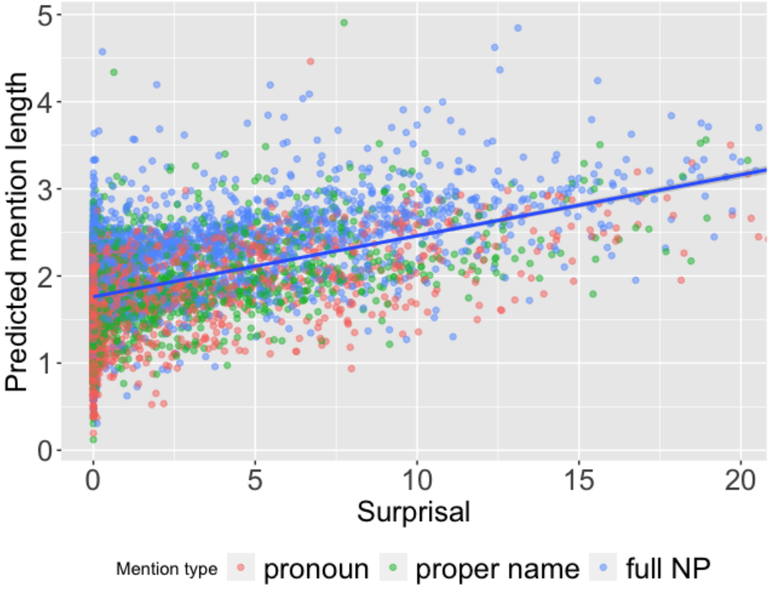


Figure 6.5: Trend between surprisal and predicted mention length by the linear regression model (Table 6.5: shallow linguistic features + surprisal), visualized by adding a smoothing line comparing only the outcome with the variable surprisal.

6.6 Summary and Discussion

In this chapter, I presented a study of the relation between referent predictability and mention form using computational estimates of the former. In the following, I discuss the implications and contributions of our findings for linguistic research. I then discuss strengths and challenges of the computational methodology we employed to obtain estimates of referent predictability.

6.6.1 Referent Predictability and Mention Form

The linguistic hypothesis we addressed in our study concerned whether speakers attempt to strike a balance between clarity and cost when producing referring expressions (Tily and Piantadosi, 2009). In particular, do speakers tend to use less informative, shorter expressions like third-person pronouns in contexts that are highly predictive of the referent? Conversely, do they reserve the more explicit expressions, like descriptions or names, for when the information in the context needs to be compensated?

When comparing referent predictability and the syntactic type of a mention, third-person pronouns are the category most consistently associated with low referent surprisal (i.e., high predictability; Figure 6.4). This constitutes evidence that ambiguous expressions are used in contexts that already contribute much information about the intended meaning of the expression. When looking at the role of referent surprisal jointly with certain linguistic features, we found that both contribute relevant information when predicting a mention type. Referent surprisal was predictive of the pronoun vs. full NPs distinction, though not of the pronoun vs. proper names distinction.

The linguistic features we took into account in our analyses (e.g., how recently a referent was mentioned, or whether it was mentioned last as the subject) are often taken to affect pronominalization (Ariel, 1990; Arnold, 1998). This effect was corroborated also by the results of our analyses. That referent surprisal contributes information on top of these features suggests that reference production is affected by factors that are accounted for by predictability but not by the linguistic features. The system we used to estimate predictability learned the aspects that are relevant to guess a referent from its context based on patterns in the data. Besides superficial features like recency, it could attend to, for instance, semantico-pragmatic factors like verb semantics or rhetorical relations (Fukumura and Van Gompel, 2010; Kehler and Rohde, 2013; Rosa and Arnold, 2017). An analysis of the type of cues that the system effectively leverages for its computations would help to clarify what information predictability contributes on top of the linguistic features (see next subsection). But also, the linguistic features contributed information on top of referent surprisal. This suggests that reference production cannot be explained by predictability alone, but rather as a complex interaction of factors.

Looking at the relation between referent predictability and mention length provided further insights, in particular in connection to the role of production cost. We found that longer expressions are associated with contexts where the intended referent is less predictable. Under the assumption that longer expressions provide more information, this suggests that speakers make the effort of producing a more costly expression when its meaning would be otherwise hard to infer. Speakers would thus be willing to compromise cost to attain clarity and make their expressions interpretable by the addressee. By contrast, short expressions are used when the context is predictive of their referent. As pronouns are consistently one-word mentions, this result is in line with the previous finding that pronouns tend to occur in predictive contexts. But the same trend is found across non-pronominal mentions only. This indicates that, across mention types, a referent's predictability affects the cost of the produced expression. This could explain the aforementioned lack of distinction between names and pronouns with surprisal: both a short proper name and a pronoun (e.g., "Tom" vs. "he") could be used for a predictable referent. Again, when jointly taking into account linguistic features, referent surprisal still contributes relevant information, but the form chosen to refer to an entity is best predicted as the result of multiple factors.

Our results on both mention type and length can be reconciled by positing the following explanation. In reference production, a speaker takes into account clarity to be collaborative with the addressee and ease the correct identification of its referent (Grice, 1975). This leads to choosing an expression whose ambiguity can be resolved in the context of what they said (and the content they plan to mention next). The speaker, however, also aims to reduce the cost of their expression, and not to convey information that is already predictable from the context (Zipf, 1949; Jaeger, 2010). This leads to saving longer (i.e., more costly) expressions – which tend to convey more information – for when the referent is not otherwise identifiable. Whenever the context is instead predictive enough, they can safely pick a pronoun, but they may also go for a more explicit expression, especially if short. This may cause redundancies between the context and the expression, contrary to what a fully efficient communication strategy would predict: why? First, being overinformative makes the utterance robust to noise: it may be welcome if not requiring much effort. Also, in referential choice, speakers may be guided by other factors –grammatical or stylistic aspects – beyond clarity and cost. Discourse factors like recency or syntactic prominence may constrain referential form to a stronger degree than their effect on predictability. For instance, the preference for pronominalizing entities mentioned in subject position was found to override the role of semantico-pragmatic aspects on reference production (Fukumura and Van Gompel, 2010; Kehler and Rohde, 2013). This connects to the open question about why certain factors affect referent predictability and production to different strengths (Arnold and Zerkle, 2019), and the extent to which the notions of predictability and salience/accessibility can be considered to overlap (Zarcone et al., 2016).

Our findings on mention type are aligned with those reported on the smaller-scale study, relying on human judgments, by Tily and Piantadosi (2009). The fact that we could extend their conclusions on a bigger and more diverse dataset speaks to the advantages of using computational estimates of referent predictability. Like Tily and Piantadosi (2009), our results are however not aligned with those reported by Modi et al. (2017), which did not find an effect of referent predictability on mention form. This may be due to the different types of data considered in the studies: Modi et al. used narrative stories centered around typical scenarios, where participants may generally be more predictable than entities in, e.g., a news text.

6.6.2 Methodological Considerations

Using computational estimates of referential expectations allowed us to scale and expedite analyses on the relation between referent predictability and mention form. We obtained these estimates by implementing a coreference resolution system that can output a probability distribution over the antecedents of an unknown mention. Before deploying our model for the analyses, we assessed the quality of its predictions. Our model's estimates of referent predictability, at least when obtained in the same setup the model was trained on, match human expectations to a satisfactory extent. This ensured that the method was reliable enough for subsequent analyses. Still, in future research, a more extensive evaluation of the model behavior would be beneficial.

First, Modi et al. (2017)'s cloze task data was not optimal for our purposes: It is not diverse in the type of texts, and it does not consider the full referring expressions (but only the mention's syntactic head). Moreover, the judgments were based only on the previous context of a mention, while the notion of predictability we adopted was not restricted to anticipation. In future work, one could collect and compare referent guesses in setups varying in the amount of context (left vs. right context). A new dataset improving in these aspects would be a useful resource, both for model evaluation and generally for linguistic analysis.

Another way in which our model of referent predictability could be further studied is by analyzing the type of cues it is sensitive to, which – recall – were autonomously learned from the data. A promising research direction is to deploy the model on targeted evaluations focusing on specific factors. We could use items manipulating specific factors (as in psycholinguistic experiments; e.g., Kehler and Rohde 2013; Fukumura and Van Gompel 2010), like the syntactic position of the last mention, or verbs with contrasting referent biases. Analogous analyses were carried out, for instance, on language models (Davis and van Schijndel, 2020). This could provide a further assessment of the quality of the system (compared to existing evidence on how humans behave). Moreover, knowing which factors are considered in the computation of referent predictability could help clarify its interaction with the linguistic features considered in our analyses.

The estimates we used in our study were based on access to both the previous and following context of an expression, but most works focus on referent predictability based only on the previous context. We motivated this choice in Section 6.2.2, out of both theoretical arguments and practical motivations. However, an avenue for future research is that of more precisely and flexibly control the amount of context that is given to a model for the prediction. For instance, we could compare results using only the left context of a mention to those with also some portion of the right context, in order to assess if and in which amount the following context plays a role on mention form. However, with the current system, focusing only on the previous context of a mention is inefficient, and also, in practice, leads to poor behavior as the model was not trained in this setup (Section 6.4). In future work, solutions to this issue could be explored, such as keeping the same architecture but relying on unidirectional LMs (as those used in Chapter 5).

Independently of ways in which our current approach could be improved, the general methodology we put forward to estimate referent predictability – using NLP techniques – can be a useful tool for linguists. In particular, it lifts the requirement of having to collect a large number of human judgments to run large-scale studies. While a model may not be a fully accurate approximation of human expectations, the potential noise introduced can be considered counterbalanced by the scale at which the observations can now be drawn. Our methodology could be applied for analogous studies on more languages than English, allowing for a cross-linguistic analysis of the relation between referent predictability and mention form. This was, for instance, the approach of Pimentel et al. (2020a) using LMs to bring evidence across languages for a balance between ambiguity and contextual informativeness at the lexical level. When focusing on reference, there is however a bottleneck in the number of languages we can test our hypotheses on, due to the requirement of sufficiently large corpora annotated for coreference (a large portion has to be reserved for training the coreference model). Still, there exist languages besides English with such resources; for instance, Arabic and Chinese are included in OntoNotes v5.0 (Weischedel et al., 2013), the resource used in our analysis. Finally, our method could be useful to explore other topics beyond reference production, such as the effect of referent predictability on processing during the resolution of referential ambiguities (McDonald and MacWhinney, 1995).

Chapter 7

CONCLUSION

This thesis reported a comprehensive study of linguistic ambiguity, which is relevant to linguistics, cognitive science, and NLP. I investigated linguistic ambiguity in the challenges it poses to its accounts and to artificial systems learning to process language. In my experiments, I considered different ways in which ambiguities can surface in language – syntactic, lexical, and referential ambiguities – focusing on the English language. Deep learning was adopted as the main research framework. On the one hand, I studied the way neural language models process ambiguities (chapters 3 and 5): To which degree are they sensitive to disambiguating cues in the context? What default preferences over interpretations do these models have, and what is the effect of these biases? On the other hand, I used deep learning models as a tool for linguistic analyses. I investigated the interaction between the information contributed by an expression and its context, by approximating these sources of information using computational models. This allowed studying interpretation and production strategies relevant to ambiguity (chapters 4 and 6).

The contributions of the thesis can be classified into *explanatory* and *methodological*: In the first case, they offer insights into the way either humans or artificial systems handle ambiguities. In the second case, they introduce novel techniques to either analyze computational models or use them to facilitate the testing of linguistic hypotheses. For each set of experiments reported, I have discussed their implications from both perspectives – explanatory and methodological – at the end of the respective chapters. In this final chapter, I consider the various findings to identify common themes and general open questions for future research.

Let me first start with the explanatory contributions. I started my thesis with a metaphor on ambiguity resolution (Figure 1.1), envisioning it as the result of balancing pans on a scale: Each pan – representing an interpretation – is loaded with weights by the expression and its context. The interpretation settles for the pan receiving more weight. The findings from my thesis corroborate this general view on ambiguity resolution, highlighting the importance of taking into account the strength to which an interpretation 1) is in general associated with the expression, and 2) is relevant to the context.

Two of the studies in this thesis (chapters 3 and 4) were concerned with lexical ambiguity resolution, both in terms of the mechanisms learned by a neural language model and those that can be hypothesized to explain human data. The findings in both studies suggest that lexical ambiguity resolution can be modeled quite successfully when starting from a lexical representation that is underspecified and frequency-biased (encoding more saliently the more frequent senses) and modulating it based on what information is contextually relevant. This idea is in line with both processing theories of word meaning access (Duffy et al., 1988; Rodd, 2020) and lexical pragmatic accounts such as relevance theory (Wilson and Carston, 2007). This result was found when explicitly formalizing and testing this mechanism with a computational model of word interpretation. This put together the lexical information from a word embedding with information that is contextually expected. But also, the neural language model BERT (Devlin et al., 2019), whose lexical disambiguation strategy is not a priori known, seems to rely on a somewhat similar process of contextualization, with contextual expectations acquiring a stronger role as the input word embedding is processed through the hidden layers.

However, our findings support a view where the degree of reliance on these two sources of information – the lexicon and context – appears to vary from situation to situation, likely to counterbalance effects of sense dominance (a bias towards the most frequent senses), or erroneous or vague contextual expectations. Some qualitative evidence of this was found in Chapter 3 and was then more directly investigated in Chapter 4. This finding opens questions and avenues for future research, both for the study of language and its modeling in NLP. First, how the focus that is put on the lexically encoded information and contextual expectations gets adapted during interpretation requires further investigation. In particular, what factors does it depend on? Our framework could provide a useful tool to investigate this, by looking at what interplay between the expectations and the lexicon better model different cases. But also it would be interesting to extend and deploy this type of framework on ambiguities beyond the word-level. Moreover, deep learning models, though already quite successful at ambiguity resolution, may benefit from taking into account this dynamic aspect of interpretation, at the level of architectural choices or inductive biases. This might in particular help in the challenging retrieval of infrequent, or even novel, senses, which deviate from the most typical usages of a word in the training corpus.

The interaction between the information in an expression and its context was found to affect not only comprehension but also production. The evidence in Chapter 6 suggests that the more a context makes a referent predictable, the more a speaker tends to reduce the informativeness of the expression used to refer to it (e.g., a pronoun vs. a description) – and vice-versa. This effect interacts with other factors (e.g., recency, syntactic position), and likely does not explain reference production alone. However, the findings are in line with the idea that speakers balance the ambiguity potential of an expression with the information that can be inferred in context. Such a mechanism can apply beyond the production of referring expressions, to explain, for instance, how lexically ambiguous words get avoided in contexts that do not permit their disambiguation (Ferreira et al., 2005; Ferreira, 2008; Pimentel et al., 2020a).

The contributions reported so far concern hypotheses about how humans handle ambiguities. Some of the findings of my thesis – chapters 3 and 5 — focus instead on the way neural language models process ambiguities. Overall, my results confirm that these systems have good disambiguation skills, though with some margin for improvement. Both in lexical and syntactic ambiguity resolution, the models acquire default biases over interpretations. For lexical ambiguity, this is directly encoded in the word embedding passed as input, and used in the hidden layers as starting point to infer the contextually relevant sense. Indeed, contextual word information, reflecting the situation-specific sense of a word, can typically be retrieved from hidden states. However, my experiments indicate that, at the same time, the underspecified lexical information passed as input remains to some degree encoded in the hidden states. In Section 3.5 I proposed some hypotheses about why this behavior emerges in a model, which however remains an open question. For syntactic ambiguity, I estimated the degree of preference towards interpretations looking at the model's output behavior (generating text). The results indicate that a language model can consider simultaneously viable two interpretations, but adjust its degree of preference for each depending on instances of a construction. This suggests that both syntactic and lexico-semantic factors (as opposed to just syntactic principles) are taken into account by the language model in the resolution of the ambiguity, analogously to mechanisms posited for human processing (MacDonald et al., 1994; Garnsey et al., 1997). However, more experiments are required to find out what aspects the model is sensitive to.

The aforementioned observations were made possible by a set of methodologies introduced in this thesis. A first group consists of techniques to use deep learning models as a tool for linguistic analysis; in particular, to obtain proxies for contextual expectations. In Chapter 4, I represented expectations about word content as a high-dimensional vector in the space of lexical representations (i.e., word embeddings). This vector can be extracted from a language model directly, at an intermediate computation step. This method was used to test the effect of expectations from the discourse context on lexical disambiguation. In Chapter 6, I modified the training setup of a coreference resolution architecture so that it estimates referent predictability based on context only. This enabled us to investigate the effect of contextual expectations on the choice of a referring expression. Both of these methods extend the toolbox of linguists with computational data-driven techniques, which I hope will inspire further studies aimed at empirically grounding linguistic hypotheses.

The methodological contributions of this thesis also include techniques to analyze the inner workings of deep learning models; in particular, their processing of ambiguity.

In Chapter 3, I used auxiliary supervised tasks to extract the word information encoded in hidden states after processing a word and its context (or just its context; *processing* vs. *predictive* hidden states). This leveraged a representation learning objective and the use of lexical substitution annotations. In Chapter 5 I instead introduced a type of behavioral analysis, focusing on the output predictions as opposed to the internal representation. I probed the expectations of a language model over an unfolding sentence using text generation, using completions to estimate the probability of an interpretation of the input. While this analysis was applied to syntactic ambiguities, it could be applied beyond this phenomenon in future work; for instance, to study the resolution of referential or lexico-semantic ambiguities. Both methodologies proved to be useful tools to clarify the dynamics of neural language models. They however also harbor some weaknesses, as discussed in Section 3.5 and 5.6. This suggests some cautiousness in the conclusions, and also highlights the importance of more research on ways in which we can uncover the dynamics of "black-box" deep learning models.

To conclude, the research presented in this thesis generally pursued two high-level goals: to find out how deep learning models process language, and how we can use them to investigate how humans do it. I believe that the second goal cannot be achieved without advancing also towards the first. Deep learning models led to great improvements in language technologies, suggesting that they may also help the work of linguists and cognitive scientists by, for instance, providing the tools to automatically estimate information. We however cannot establish whether a certain method is suitable for our goals without taking a closer look at the representations or estimates it introduces. If using deep learning models, we need to take into account the implications of architectural or training decisions, or the results of analyses on what these models encode and how: What assumptions am I introducing? Am I obtaining an adequately reliable proxy? But, with the appropriate caveats, deep learning models offer much potential. I hope that the ideas put forward in this thesis will inspire new applications of computational data-driven techniques to enhance our understanding of how language works.

Appendix A

CHAPTER 3

A.1 biLSTM Language Model

The language model was trained to optimize the log-likelihood of a target word given its surrounding context, with stochastic gradient descent for 20 epochs with decaying learning rate using Adam optimizer Kingma and Ba (2014). The initial learning rate was 0.0005 for a batch size of 32. Dropout was set to 0.2 and applied to the input embedding, and the outputs of the LSTM layers. At training time, the text data is fed to the model in sequences of 100 tokens. On *test* data, the model achieved a language modeling perplexity of 18.06.

A.2 Diagnostic Models

Separate diagnostic models are trained for each combination of language model (biLSTM or BERT), task (**lex** or **ctxt**) and input representation type (a processing or predictive hidden state at a given layer). Each model consists of a non-linear transformation with $tahn$ non-linearity, trained using Cosine Embedding Loss (PyTorch 0.4, Paszke et al., 2017) and Adam optimizer, with early stopping based on validation loss. We carried out hyperparameter search based on validation loss for each of the model types in order to set batch size and initial learning rate. We report the final settings for each combination of input and task in Table A.1.

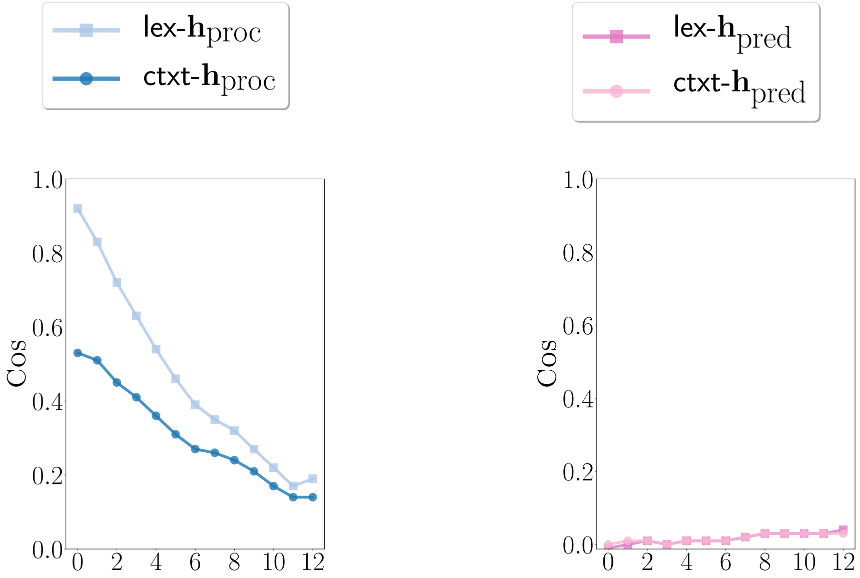
A.3 Hidden States and Word Representations

Figure A.1 displays the average cosine of hidden states – processing and predictive – to the representations of word information used as targets by the diagnostic models. **lex** is the context-invariant representation of lexical information – the word embedding, while

	biLSTM			
input	**lex**		**ctxt**	
	pred	proc	pred	proc
$\mathbf{h}_t^1$	(64, 5e-05)	(64, 0.0005)	(32, 0.0001)	(64, 0.0005)
$\mathbf{h}_t^2$	(64, 5e-05)	(64, 0.0005)	(16, 0.0001)	(16, 0.0005)
$\mathbf{h}_t^3$	(32, 5e-05)	(32,0.001)	(32, 5e-05)	(64, 0.0001)

	BERT			
input	**lex**		**ctxt**	
	pred	proc	pred	proc
$\mathbf{h}_t^0$	(128, 5e-05)	(64, 5e-05)	(128, 0.0001)	(64, 5e-05)
$\mathbf{h}_t^1$	(128, 5e-05)	(32, 0.0001)	(64, 5e-05)	(16, 5e-05)
$\mathbf{h}_t^2$	(128, 5e-05)	(128, 0.0005)	(64, 5e-05)	(128, 0.0005)
$\mathbf{h}_t^3$	(128, 5e-05)	(128,0.0005)	(64, 5e-05)	(16, 0.0001)
$\mathbf{h}_t^4$	(128, 5e-05)	(16, 0.0001)	(32, 5e-05)	(16, 0.0001)
$\mathbf{h}_t^5$	(128, 0.0001)	(64, 0.0005)	(64, 5e-05)	(16,5e-05)
$\mathbf{h}_t^6$	(128, 5e-05)	(64, 0.0005)	(64, 5e-05)	(16, 0.0001)
$\mathbf{h}_t^7$	(128, 0.0001)	(64, 0.0005)	(64, 5e-05)	(16, 5e-05)
$\mathbf{h}_t^8$	(32, 5e-05)	(128, 0.0005)	(16, 5e-05)	(16, 0.0001)
$\mathbf{h}_t^9$	(64, 5e-05)	(128,0.0005)	(32, 5e-05)	(64, 0.0001)
$\mathbf{h}_t^{10}$	(16, 5e-05)	(16, 0.0001)	(16, 5e-05)	(16, 0.0001)
$\mathbf{h}_t^{11}$	(32, 5e-05)	(64, 0.0005)	(16, 5e-05)	(16, 5e-05)
$\mathbf{h}_t^{12}$	(16, 0.0001)	(16,0.0001)	(16, 0.0001)	(64, 0.0005)

Table A.1: Hyperparameter settings in the diagnostic models (batch size, initial learning rate)



(a) Processing hidden states

(b) Predictive hidden states

Figure A.1: Average cosine of hidden states (processing and predictive) to lexical and contextual representations (**lex** and **ctxt**, respectively).

ctxt is modulated to reflect the context-specific interpretation of a word. As it can be seen, for processing hidden states the cosine to word representations is very high in the initial layers, especially to the lexical representation. Recall that the word embedding is passed as input for processing to these layers: what these results suggest is that the hidden state retains a high similarity to it initially. The cosine to the input word embedding then dramatically decreases as more transformations are applied across the following layers. This has the effect of shifting hidden states away from the representations that need to be retrieved by a diagnostic model. In particular, there is a larger cosine margin to recuperate in the latest layers. This does not entail that these states do not contain relevant word information but suggests that retrieving it from the hidden states and mapping it to the correct subspace might be a more difficult task, to start with.

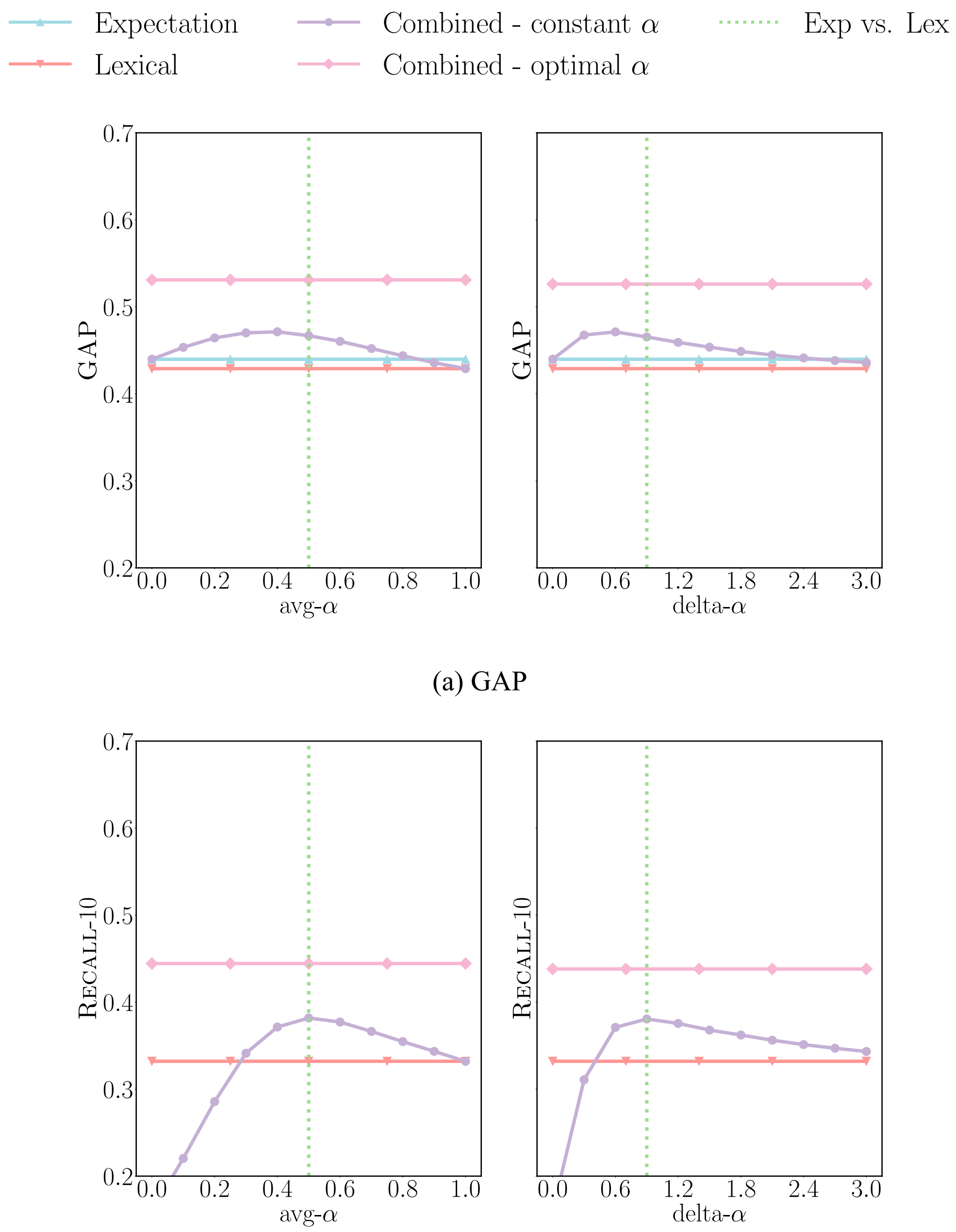
By contrast, processing hidden states do not exhibit this effect. There is a slight increase in cosine moving towards the layers closer to the output, but generally, they are uniformly very far away from the word subspace.

Appendix B

CHAPTER 4

B.1 Further Results

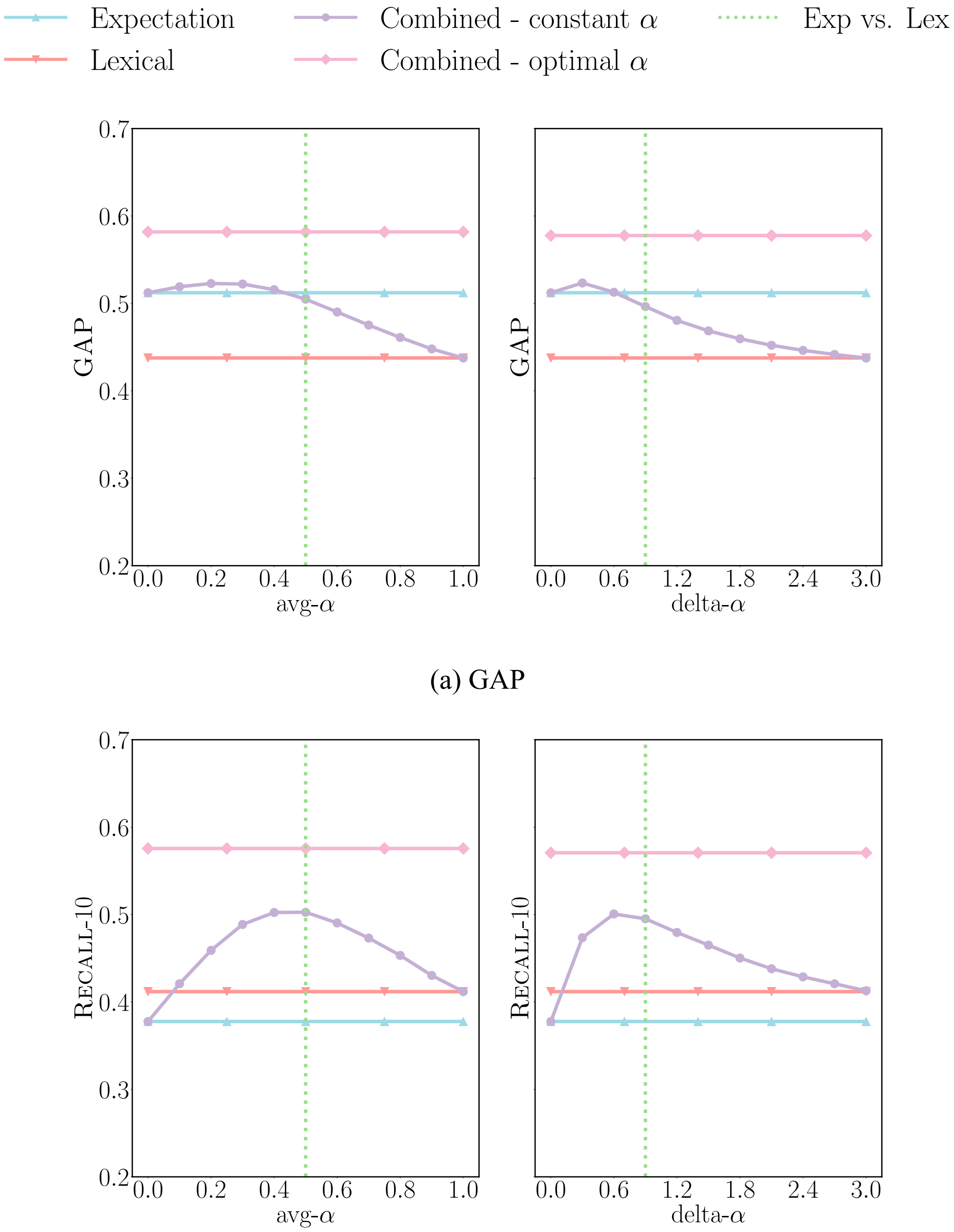
Figure B.1 and B.2 display the LexSub results obtained with different α values on *dev* data, for the biLSTM and BERT-base respectively (the plots relative to BERT-large are reported in the body of the thesis). While the magnitude of the scores and the α value where they achieve their peak vary across models, overall the trends are comparable. In particular, increasing the constant α initially benefits the yielded interpretations, to then becoming detrimental above a certain threshold. Moreover, selecting for each datapoint its optimal α leads to accounting best for the substitution judgments.



(a) GAP

(b) Recall-10

Figure B.1: Results on *dev* data for the biLSTM. Dotted lines mark whether a given α yields a $\mathbf{i}_{v,c}$ that is closer to $\mathbf{e}_c$ (to the line's left) than to $\mathbf{l}_v$ (to its right).



(a) GAP

(b) Recall-10

Figure B.2: Results on *dev* data for BERT-base. Dotted lines mark whether a given α yields a $\mathbf{i}_{v,c}$ that is closer to $\mathbf{e}_c$ (to the line's left) than to $\mathbf{l}_v$ (to its right).

Appendix C

CHAPTER 5

C.1 Generation: Further Details

To generate completions from the prompt, we apply stochastic decoding as described in Section 5.3 Due to the different types of vocabulary used by the two language models – word-level vs. with both word and subword units (Byte-Pair-Encoding), some differences are introduced to generate sentence completions from the models. From a prompt, we generate 30 and 50 tokens for the LSMT and GPT2, respectively. This is because GPT2 can generate subwords, and thus require more steps on average to reach the end of a sentence. For the analyses, we discard GPT2 completions where the last word of the prompt is followed by a subword, thus changing its identity (e.g., from "suit" to "suitable"). For the LSTM we penalize the generation of the unknown-word symbol (used when a word is not covered by the vocabulary), reducing its output score by a factor of 10^{16}. This ensures that sampling this symbol is unlikely.

A minority of words in NP/Z and Noun/Verb prompts were not in the vocabulary of the LSTM. To avoid having to use the unknown-word symbol and to make sure that the model has access to all the relevant information in the prompt, we replace these words with equivalent ones that do not substantially affect the meaning of the sentence (e.g., "jogger" → "runner"). We use these modified prompts for the experiments on both language models.

C.2 Diversity of Generated Sentences

We analyze the diversity of the completions generated for each prompt. Completions were rarely completely identical: the average proportion of unique completions in a sample (pooling together all prompt types) is at least 98% for all ambiguity types and LMs. Of course, two completions can be very similar, though not identical. To measure to extent of this phenomenon, we measure the lexical overlap across completions, focusing on unigrams and bigrams. We calculate individual Self-BLEU scores of each completion with respect to the others generated for the same prompt. Average unigram scores tend to be much higher than the bigram ones across ambiguity types and LMs (the former in the range .68-.71, while the latter .18-.25). This shows that individual words are often repeated across completions, but not so frequently in the same order.

C.3 Classifying Completions

I here describe the rules employed to classify completions generated by a language model based on the inferred syntactic interpretation of the prompt.

C.3.1 NP/S and NP/Z Sentences

It is possible to distinguish between the NP and S interpretations and between the NP and Z interpretation through the syntactic role of the head of the noun phrase that contains the locus of ambiguity. We turn this principle into concrete rules to classify a sentence based on the dependency labels assigned to its tokens by a dependency parser.

The rules are applied recursively:

- In the base case, we check if the predicted label of a given token is that of direct object or subject. If it is a subject, we return S for NP/S and Z for NP/Z.
- If the label for the given token is neither subject nor direct object, we consider whether it could be part of a complex NP. If the predicted label is that of a modifier or possessive, we apply the function to the following tokens to identify the head of the NP and determine whether it is a subject or direct object. If this scenario does not apply, we return *other*.

The most common parser error involves a failure to detect S or Z cases, labeling the locus as direct object when followed by a finite verb. This parse is not grammatical, because the verb after the locus of ambiguity is left without a subject. The locus of ambiguity then needs to be the subject itself. We modify the base case rule to detect and correct these cases:

- If the given token is a direct object but it is followed by a token with a dependency label compatible with a finite verb (e.g., *root, ccomp*), we change the label to subject, and return S for NP/S and Z for NP/Z.

C.3.2 NP/Z Sentences with Disambiguation Issues

In Chapter 5 we described a subset of completions to Post-locus cue completions of NP/Z prompts that indicate that the LMs have not fully adapted to the Z interpretation. To identify this behavior and quantify it across the data, we use patterns in the dependency labels predicted for a sentence. In particular, the following condition:

- The post-locus cue is not recognized as the main verb;
- A comma is placed between the post-locus cue verb and the main verb;
- The comma is not followed by a conjunction.

C.3.3 The Noun/Verb Ambiguity

For the Noun/Verb ambiguity, we classify completions based the part-of-speech tag predicted by the parser for the locus of ambiguity (*NN, NNS* etc. → Noun; *VB, MD* etc. → Verb). Errors in the part-of-speech tags predicted by the parser tend to cause incorrect dependency labels as well; as such, we do not rely on the dependency labels for this ambiguity.

Appendix D

CHAPTER 6

D.1 Masking mentions: details

For simplicity, both in training and evaluation, we never mask mentions which are embedded in another mention (e.g., “the bride” in “the mother of the bride”), since that would cover information relevant to the larger mention. In case we mask a mention that includes another mention, we discard the latter from the set of mentions for which to compute a prediction. When evaluating antecedent prediction, we skip the first mention in a document as this is a trivial prediction (no antecedent).

D.2 Coreference resolution on OntoNotes

Table D.1 reports MUC, B^3 and CEAF scores, whose F1 average we report (CoNLL-2012 score), for the Coref and Coref_{M}, both with predicted and gold mention boundaries. The results are overall comparable between the two systems across all metrics.

In Chapter 6, I report the results of antecedent prediction by mention type for Coref and Coref_{M} using gold mention boundaries. I here report the same results for different setups and models. First, the results of Coref and Coref_{M} by mention type using predicted mention boundaries are displayed in Figure D.2 and D.1. The trends across mention types are the same as with gold mentions (Table 6.3), but the results are generally worse, plausibly due to errors introduced by the additional challenge of mention detection. In Figure D.3 I report the results of a variant of Coref_{M} – $\text{Coref}_{\text{M}}{}^3$ – where, both during training and deployment, a mention is masked substituting it with a sequence of three `[MASK]` tokens, instead of just one. This is to verify whether the use of a single token is responsible for the quality of predictions on one-token mentions, and in particular pronouns. The results show that this is not the case, as the trends found with Coref_{M} are the same as those with $\text{Coref}_{\text{M}}{}^3$: when a third-person pronoun is used the antecedent is still easier to predict than when a proper name is used, and even less than a full NP.

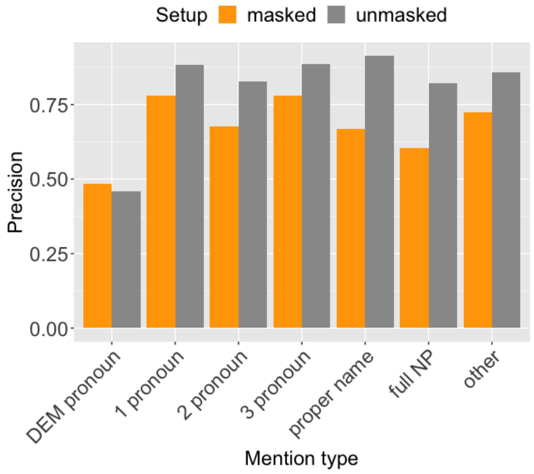


Figure D.1: Antecedent precision scores with predicted mentions of the model Coref$_{\text{M}}$ across different mention types, for both masked and unmasked mentions.

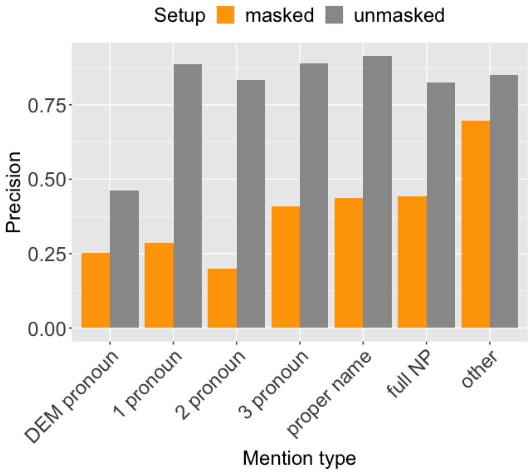


Figure D.2: Antecedent precision scores with predicted mentions of the model Coref across different mention types, for both masked and unmasked mentions.

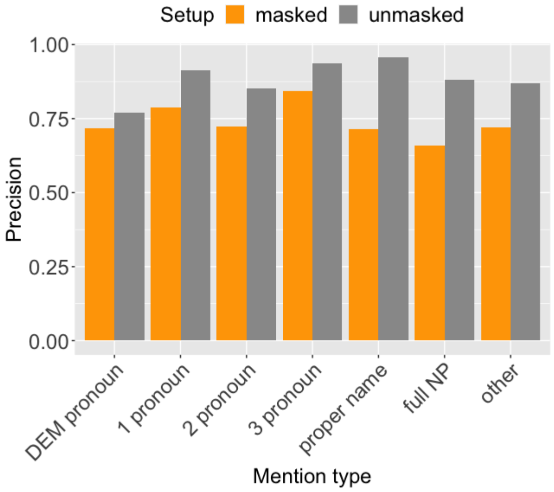


Figure D.3: Antecedent precision scores with gold mentions of the model Coref$_{\text{M}}$^{3} across different mention types, for both masked and unmasked mentions.

model	mentions	MUC			B^3			CEAF			Avg		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Coref	predicted	.84	.83	.84	.76	.75	.76	.75	.71	.73	.78	.77	.77
	gold	.95	.91	.93	.87	.86	.86	.92	.77	.84	.91	.85	.88
Coref_{M}	predicted	.84	.83	.83	.76	.75	.75	.74	.71	.73	.78	.76	.77
	gold	.95	.93	.94	.86	.88	.87	.92	.78	.85	.91	.86	.88

Table D.1: Results on OntoNotes *test* data in document-level coreference resolution (only with unmasked mentions); P, R, F1 = precision, recall, F1 scores.

	Proper name			Full NP				
	β	s.e.	z	β	s.e.	z	LR_{χ^2}	dfp_{χ^2}
Intercept	-.63	.03	-23.77	-.26	.02	-10.93		
surprisal	.31	.03	9.56	.47	.03	16.44	330.28	2 *
Intercept	-.24	.07	-3.60	.04	.07	0.58		
distance	3.13	.12	25.40	3.10	.12	25.23	1466.81	2 *
frequency	.09	.03	3.11	-.13	.03	-3.78	37.35	2 *
antecedent = previous subject	-1.31	.09	-13.91	-1.10	.08	-13.70	362.02	2 *
mention = subject	.07	.07	1.00	-0.50	.06	-7.71	82.57	2 *
antecedent type = proper name	1.78	.08	22.83	.41	.09	4.62	1723.67	2 *
full NP	-.17	.08	-2.16	1.18	.06	18.13		
surprisal	.05	.04	1.46	.23	.03	7.80	75.18	2 *

Table D.2: Multinomial logit model with shallow linguistic features and surprisal predicting mention type (baseline level is "pronoun"). Coefficients which have an absolute *z* score lower than 2 are not significant. * : $p_{\chi^2} < 0.001$.

D.3 Predicting Mention Form: More Results

Table D.2 and D.3 add results from likelihood-ratio chi-square tests and F-tests, respectively. The two tables complement the information in Table 6.4 and 6.5. All variables are tested to significantly improve goodness-of-fit to the data, except the feature " mention = subject" in predicting mention length (number of tokens).

Table D.4 shows two linear regression models predicting mention length quantified in terms of the number of tokens for each non-pronominal mention (proper noun, full NP). We find that longer non-pronominal expressions are favored when surprisal increases, indicating that the trends reported when including also pronominal expressions were not simply due to the distinction between pronouns and non-pronouns. Rather a predictive relation between referent surprisal and mention length seems to apply in a general way, across all mention types.

		β	s.e.	t	p_t	F	df	p_F
Intercept		1.87	.02	80.75	*			
surprisal		.25	.02	10.74	*	115.32	1	*
Intercept		1.81	.05	40.08	*			
distance		0.17	0.02	7.08	*	50.08	1	*
frequency		-.13	.02	-5.37	*	28.82	1	*
antecedent = previous subject		-.51	.06	-8.46	*	71.59	1	*
mention = subject		.04	.05	.77	.44	.60	1	0.44
antecedent type =	proper noun	-.21	.06	-3.21	.0013	59.32	2	*
	full NP	.42	.06	7.51	*			
surprisal		.17	.02	7.37	*	54.37	1	*

Table D.3: Two linear regression models predicting mention length (number of tokens) of the masked mention, based on 1) surprisal alone and 2) shallow linguistic features + surprisal. All predictors were tested to improve goodness-of-fit to the data except ”grammatical function of mention”. $*: p < 0.001$.

		β	s.e.	t	p_t	F	df	p_F
Intercept		2.53	.04	64.24	-			
surprisal		.18	.04	5.10	*	26.00	1	*
Intercept		2.69	.09	30.72	-			
distance		-.02	.03	-0.71	.48	.51	1	.48
frequency		-.31	.05	-6.87	*	47.17	1	*
antecedent = previous subject		-.14	.15	-.93	.35	.87	1	.35
mention = subject		.10	.09	1.05	.29	1.11	1	.29
antecedent type =	proper noun	-.96	.11	-8.71	*	77.26	2	*
	full NP	.16	.10	1.56	.12			
surprisal		.16	.04	4.44	*	19.74	1	*

Table D.4: Two linear regression models predicting mention length for each masked non-pronominal mention, based on 1) surprisal alone and 2) shallow linguistic features + surprisal. $*: p < 0.001$

Bibliography

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of 5th ICLR International Conference on Learning Representations*.

Aina, L., Silberer, C., Sorodoc, I.-T., Westera, M., and Boleda, G. (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.

Alishahi, A., Chrupała, G., and Linzen, T. (2019). Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*, 25(4):543–557.

Apresjan, J. D. (1974). Regular Polysemy. *Linguistics*, 12(142):5–32.

Arefyev, N., Sheludko, B., Podolskiy, A., and Panchenko, A. (2020). A Comparative Study of Lexical Substitution Approaches based on Neural Language Models. *arXiv preprint arXiv:2006.00031*.

Ariel, M. (1990). *Accessing Noun-Phrase antecedents*. Routledge.

Armeni, K., Willems, R. M., and Frank, S. L. (2017). Probabilistic language models in cognitive neuroscience: Promises and pitfalls. *Neuroscience & Biobehavioral Reviews*, 83:579–588.

Arnold, J. E. (1998). *Reference form and discourse patterns*. PhD thesis, Stanford University.

Arnold, J. E. (2001). The Effect of Thematic Roles on Pronoun Use and Frequency of Reference Continuation. *Discourse Processes*, 31(2):137–162.

Arnold, J. E. and Zerkle, S. A. (2019). Why do people produce pronouns? Pragmatic selection vs. rational models. *Language, Cognition and Neuroscience*, 34(9):1152–1175.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Asher, N. (2011). *Lexical Meaning in Context: A Web of Words*. Cambridge University Press.

Asher, N., Abrusan, M., and Van de Cruys, T. (2017). Types, meanings and co-composition in lexical semantics. In *Modern Perspectives in Type-Theoretical Semantics*, pages 135–161. Springer.

Asher, N. and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Aylett, M. and Turk, A. (2004). The Smooth Signal Redundancy Hypothesis: A Functional Explanation for Relationships between Redundancy, Prosodic Prominence, and Duration in Spontaneous Speech. *Language and Speech*, 47(1):31–56.

Baroni, M. (2021). On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. *arXiv preprint arXiv:2106.08694*.

Belinkov, Y. and Glass, J. (2019). Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Bernardi, R., Boleda, G., Fernández, R., and Paperno, D. (2015). Distributional Semantics in Use. In *Proceedings of the EMNLP 2015 Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 95–101, Lisboa, Portugal. Association for Computational Linguistics.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In Hayes, J. R., editor, *Cognition and the development of language*, pages 279–362. New York: John Wiley.

Blott, L. M., Rodd, J. M., Ferreira, F., and Warren, J. E. (2020). Recovery from misinterpretations during online sentence processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 47(6):968–997.

Boleda, G. (2020). Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

Camacho-Collados, J. and Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*.

Carston, R. (2012). Word meaning and concept expressed. *The Linguistic Review*, 29(4):607–623.

Carston, R. (2019). Ad hoc concepts, polysemy and the lexicon. In *Relevance, Pragmatics and Interpretation*, pages 150–162. Cambridge University Press.

Carston, R. (2021). Polysemy: Pragmatics and sense conventions. *Mind & Language*, 36(1):108–133.

Chafe, W. (1996). Inferring identifiability and accessibility. *Pragmatics & Beyond New Series: Reference and Referent Accessibility*, 38:37–46.

Chomsky, N. (2002). An interview on minimalism. In *On Nature and Language*. Cambridge University Press.

Christianson, K., Hollingworth, A., Halliwell, J. F., and Ferreira, F. (2001). Thematic Roles Assigned along the Garden Path Linger. *Cognitive Psychology*, 42(4):368–407.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204.

Clark, E. V. and Clark, H. H. (1979). When Nouns Surface as Verbs. *Language*, pages 767–811.

Clifton Jr., C. and Staub, A. (2008). Parallelism and competition in syntactic ambiguity resolution. *Language and Linguistics Compass*, 2(2):234–250.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Cosentino, E., Baggio, G., Kontinen, J., and Werning, M. (2017). The Time-Course of Sentence Meaning Composition. N400 Effects of the Interaction between Context-Induced and Lexically Stored Affordances. *Frontiers in Psychology*, 8:813.

Cruse, D. A. (1986). *Lexical semantics*. Cambridge University Press.

Davies, C. and Arnold, J. (2019). Reference and informativeness: How context shapes referential choice. *The Oxford Handbook of Experimental Semantics and Pragmatics*.

Davis, F. and van Schijndel, M. (2020). Discourse structure interacts with reference but not syntax in neural language models. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 396–407, Online. Association for Computational Linguistics.

Deemter, K. v. (1990). Forward References in Natural Language. *Journal of Semantics*, 7(3):281–300.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dozat, T. and Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th ICLR International Conference on Learning Representations*.

Duffy, S. A., Kambe, G., and Rayner, K. (2001). The effect of prior disambiguating context on the comprehension of ambiguous words: Evidence from eye movements. In *On the consequences of meaning selection: Perspectives on resolving lexical ambiguity*, pages 27–43. American Psychological Association.

Duffy, S. A., Morris, R. K., and Rayner, K. (1988). Lexical ambiguity and fixation times in reading. *Journal of Memory and Language*, 27(4):429–446.

Elkahky, A., Webster, K., Andor, D., and Pitler, E. (2018). A challenge set and methods for noun-verb ambiguity. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2562–2572, Brussels, Belgium. Association for Computational Linguistics.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Erk, K. (2010). What is word meaning, really? (and how can distributional models help us describe it?). In *Proceedings of the 2010 workshop on geometrical models of natural language semantics*, pages 17–26. Association for Computational Linguistics.

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii. Association for Computational Linguistics.

Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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 55–65.

Ettinger, A. (2020). What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Falkum, I. L. (2015). The how and why of polysemy: A pragmatic account. *Lingua*, 157:83–99. Polysemy: Current Perspectives and Approaches.

Falkum, I. L. and Vicente, A. (2015). Polysemy: Current perspectives and approaches. *Lingua*, 157:1–16. Polysemy: Current Perspectives and Approaches.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukWac, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Ferreira, F., Bailey, K. G., and Ferraro, V. (2002). Good-Enough Representations in Language Comprehension. *Current Directions in Psychological Science*, 11(1):11–15.

Ferreira, F. and Patson, N. D. (2007). The ‘Good Enough’ Approach to Language Comprehension. *Language and Linguistics Compass*, 1(1-2):71–83.

Ferreira, V. S. (2008). Ambiguity, Accessibility, and a Division of Labor for Communicative Success. *Psychology of Learning and Motivation*, 49:209–246.

Ferreira, V. S., Slevc, L. R., and Rogers, E. S. (2005). How do speakers avoid ambiguous linguistic expressions? *Cognition*, 96(3):263–284.

Ferrer i Cancho, R. and Solé, R. V. (2003). Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences*, 100(3):788–791.

Firth, J. R. (1957). *A Synopsis of Linguistic Theory, 1930-1955*. Blackwell.

Foraker, S. and Murphy, G. L. (2012). Polysemy in sentence comprehension: Effects of meaning dominance. *Journal of Memory and Language*, 67(4):407–425.

Frank, M. C. and Goodman, N. D. (2012). Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Frank, S. L., Otten, L. J., Galli, G., and Vigliocco, G. (2013). Word surprisal predicts N400 amplitude during reading. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 878–883.

Frazier, L. (1978). *On comprehending sentences: Syntactic parsing strategies*. PhD thesis, University of Connecticut.

Frazier, L. and Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition*, 6(4):291–325.

Frazier, L. and Rayner, K. (1987). Resolution of syntactic category ambiguities: Eye movements in parsing lexically ambiguous sentences. *Journal of Memory and Language*, 26(5):505–526.

Frazier, L. and Rayner, K. (1990). Taking on semantic commitments: Processing multiple meanings vs. multiple senses. *Journal of Memory and Language*, 29(2):181–200.

Frege, G. (1892). Sense and reference. *The Philosophical Review*, 57(3):209–230.

Frisson, S. (2009). Semantic underspecification in language processing. *Language and Linguistics Compass*, 3(1):111–127.

Frisson, S. and Pickering, M. J. (1999). The processing of metonymy: Evidence from eye movements. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(6):1366.

Fukumura, K. and Van Gompel, R. P. (2010). Choosing anaphoric expressions: Do people take into account likelihood of reference? *Journal of Memory and Language*, 62(1):52–66.

Futrell, R., Wilcox, E., Morita, T., and Levy, R. (2018). RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., and Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42.

Gale, W., Church, K. W., and Yarowsky, D. (1992). Estimating Upper and Lower Bounds on the Performance of Word-Sense Disambiguation Programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Newark, Delaware, USA. Association for Computational Linguistics.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Garí Soler, A. and Apidianaki, M. (2021). Let's Play Mono-Poly: BERT Can Reveal Words' Polysemy Level and Partitionability into Senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

Garí Soler, A., Apidianaki, M., and Allauzen, A. (2019a). Word Usage Similarity Estimation with Sentence Representations and Automatic Substitutes. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 9–21.

Garí Soler, A., Cocos, A., Apidianaki, M., and Callison-Burch, C. (2019b). A Comparison of Context-sensitive Models for Lexical Substitution. In *Proceedings of the 13th International Conference on Computational Semantics (IWCS)*.

Garnsey, S., Pearlmutter, N., Myers, E., and Lotocky, M. (1997). The contributions of verb bias and plausibility to the comprehension of temporarily ambiguous sentences. *Journal of Memory and Language*, 37(1):58–93.

Giulianelli, M., Harding, J., Mohnert, F., Hupkes, D., and Zuidema, W. (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.

Givón, T. (1983). *Topic continuity in discourse*. John Benjamins Publishing Company.

Goodkind, A. and Bicknell, K. (2018). Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Goodman, N. D. and Frank, M. C. (2016). Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Grice, P. (1975). Logic and Conversation. *Syntax and Semantics 3: Speech Acts*, pages 41 – 58.

Grodner, D., Gibson, E., Argaman, V., and Babyonyshev, M. (2003). Against repair-based reanalysis in sentence comprehension. *Journal of Psycholinguistic Research*, 32(2):141–166.

Grosz, B. J., Joshi, A. K., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Gulordava, K., Aina, L., and Boleda, G. (2018a). How to represent a word and predict it, too: Improving tied architectures for language modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2936–2941, Brussels, Belgium. Association for Computational Linguistics.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018b). 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.

Gundel, J. K., Hedberg, N., and Zacharski, R. (1993). Cognitive Status and the form of Referring Expressions in Discourse. *Language*, 69:274–307.

Hale, J. (2001). A Probabilistic Earley Parser as a Psycholinguistic Model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Harris, Z. S. (1954). Distributional Structure. *Word*, 10(2-3):146–162.

Hewitt, J. and Manning, C. D. (2019). A Structural Probe for Finding Syntax in Word Representations. 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 4129–4138.

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*, 44(4):311–338.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2019). The Curious Case of Neural Text Degeneration. In *Proceedings of 7th International Conference on Learning Representations*.

Hupkes, D., Veldhoen, S., and Zuidema, W. (2018). Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Ide, N., Baker, C., Fellbaum, C., and Passonneau, R. (2010). The Manually Annotated Sub-Corpus: A Community Resource for and by the People. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 68–73, Uppsala, Sweden. Association for Computational Linguistics.

Inan, H., Khosravi, K., and Socher, R. (2017). Tying word vectors and word classifiers: A loss framework for language modeling. In *Proceedings of 5th ICLR International Conference on Learning Representations*.

Jaeger, T. F. (2010). Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61(1):23–62.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019). BERT for coreference resolution: Baselines and analysis. 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 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Jumelet, J., Zuidema, W., and Hupkes, D. (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, Hong Kong, China. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Kao, J., Bergen, L., and Goodman, N. (2014). Formalizing the Pragmatics of Metaphor Understanding. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society (CogSci 2014)*.

Kehler, A., Kertz, L., Rohde, H., and Elman, J. L. (2008). Coherence and Coreference Revisited. *Journal of Semantics*, 25(1):1–44.

Kehler, A. and Rohde, H. (2013). A probabilistic reconciliation of coherence-driven and centering-driven theories of pronoun interpretation. *Theoretical Linguistics*, 39(1-2):1–37.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klafka, J. and Ettinger, A. (2020). Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4811.

Klein, D. E. and Murphy, G. L. (2001). The representation of polysemous words. *Journal of Memory and Language*, 45(2):259–282.

Klepousniotou, E. (2002). The processing of lexical ambiguity: Homonymy and polysemy in the mental lexicon. *Brain and Language*, 81(1-3):205–223.

Kremer, G., Erk, K., Padó, S., and Thater, S. (2014). What Substitutes Tell Us - Analysis of an "All-Words" Lexical Substitution Corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden. Association for Computational Linguistics.

Kuperberg, G. R. and Jaeger, T. F. (2016). What do we mean by prediction in language comprehension? *Language, Cognition and Neuroscience*, 31(1):32–59.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (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.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Leech, G. (1992). 100 Million Words of English:The British National Corpus (BNC). *Second Language Research*, 28:1–13.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31.

Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics*.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

Levy, R. (2013). Memory and Surprisal in Human Sentence Comprehension. In van Gompel, R. P. G., editor, *Sentence Processing*, page 78–114. Hove: Psychology Press.

Levy, R. and Jaeger, T. F. (2007). Speakers optimize information density through syntactic reduction. *Advances in Neural Information Processing Systems*, 19:849.

Li, J. and Joanisse, M. F. (2021). Word Senses as Clusters of Meaning Modulations: A Computational Model of Polysemy. *Cognitive Science*, 45(4):e12955.

Linzen, T. (2019). What can linguistics and deep learning contribute to each other? Response to Pater. *Language*, 95(1):e99–e108.

Linzen, T. and Baroni, M. (2021). Syntactic Structure from Deep Learning. *Annual Review of Linguistics*, 7(1):195–212.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Loureiro, D., Rezaee, K., Pilehvar, M. T., and Camacho-Collados, J. (2021). Analysis and Evaluation of Language Models for Word Sense Disambiguation. *Computational Linguistics*, 47(2):387–443.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Ludlow, P. (2014). *Living words: Meaning underdetermination and the dynamic lexicon*. Oxford University Press.

MacDonald, M. C., Pearlmutter, N. J., and Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological review*, 101(4):676.

Manning, C. and Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Manning, C. D. (2015). Computational Linguistics and Deep Learning. *Computational Linguistics*, 41(4):701–707.

Marvin, R. and Linzen, T. (2018). Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

McCarthy, D. and Navigli, R. (2007). SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

McCarthy, D. and Navigli, R. (2009). The English lexical substitution task. *Language resources and evaluation*, 43(2):139–159.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

McDonald, J. L. and MacWhinney, B. (1995). The Time Course of Anaphor Resolution: Effects of Implicit Verb Causality and Gender. *Journal of Memory and Language*, 34(4):543–566.

McRae, K., Spivey-Knowlton, M. J., and Tanenhaus, M. K. (1998). Modeling the Influence of Thematic Fit (and Other Constraints) in On-line Sentence Comprehension. *Journal of Memory and Language*, 38(3):283–312.

Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Melamud, O., Levy, O., and Dagan, I. (2015). A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st ICLR International Conference on Learning Representations*.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of 11t Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244.

Mitchell, J. and Lapata, M. (2010). Composition in Distributional Models of Semantics. *Cognitive science*, 34(8):1388–1429.

Modi, A., Anikina, T., Ostermann, S., and Pinkal, M. (2016). InScript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3485–3493, Portorož, Slovenia. European Language Resources Association (ELRA).

Modi, A., Titov, I., Demberg, V., Sayeed, A., and Pinkal, M. (2017). Modeling Semantic Expectation: Using Script Knowledge for Referent Prediction. *Transactions of the Association for Computational Linguistics*, 5:31–44.

Nair, S., Srinivasan, M., and Meylan, S. (2020). Contextualized Word Embeddings Encode Aspects of Human-Like Word Sense Knowledge. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 129 – 141, Barcelona, Spain (Online).

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2).

Nieuwland, M. S. and Van Berkum, J. J. A. (2008). The Neurocognition of Referential Ambiguity in Language Comprehension. *Language and Linguistics Compass*, 2(4):603–630.

Orita, N., Vornov, E., Feldman, N., and Daumé III, H. (2015). Why discourse affects speakers' choice of referring expressions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1639–1649.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Pedinotti, P., Rambelli, G., Chersoni, E., Santus, E., Lenci, A., and Blache, P. (2021). Did the Cat Drink the Coffee? Challenging Transformers with Generalized Event Knowledge. In *Proceedings of the 10th Joint Conference on Lexical and Computational Semantics (*SEM 2021)*, pages 1–11.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018a). Deep Contextualized Word Representations. 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 2227–2237.

Peters, M. E., Neumann, M., Zettlemoyer, L., and Yih, W.-t. (2018b). Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Piantadosi, S. T., Tily, H., and Gibson, E. (2011). Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Piantadosi, S. T., Tily, H., and Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, 122(3).

Pickering, M. J. and Van Gompel, R. P. (2006). Syntactic parsing. In *Handbook of Psycholinguistics*, pages 455–503. Elsevier.

Pilehvar, M. T. and Camacho-Collados, J. (2019). WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. 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 1267–1273.

Pimentel, T., Hall Maudslay, R., Blasi, D., and Cotterell, R. (2020a). Speakers Fill Lexical Semantic Gaps with Context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4004–4015, Online. Association for Computational Linguistics.

Pimentel, T., Valvoda, J., Maudslay, R. H., Zmigrod, R., Williams, A., and Cotterell, R. (2020b). Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Press, O. and Wolf, L. (2017). Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163.

Pritchett, B. L. (1988). Garden Path Phenomena and the Grammatical Basis of Language Processing. *Language*, 64:539–576.

Pustejovsky, J. (1991). The generative lexicon. *Computational linguistics*, 17(4):409–441.

Pustejovsky, J. (1995). *The Generative Lexicon*. MIT Press.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Rayner, K. and Frazier, L. (1989). Selection Mechanisms in Reading Lexically Ambiguous Words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(5):779.

Recanati, F. (2004). *Literal meaning*. Cambridge University Press.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., and Kim, B. (2019). Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Reisinger, J. and Mooney, R. J. (2010). Multi-Prototype Vector-Space Models of Word Meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California. Association for Computational Linguistics.

Rodd, J. (2018). Lexical Ambiguity. *Oxford Handbook of Psycholinguistics*, pages 120–144.

Rodd, J., Gaskell, G., and Marslen-Wilson, W. (2002). Making Sense of Semantic Ambiguity: Semantic Competition in Lexical Access. *Journal of Memory and Language*, 46(2):245–266.

Rodd, J. M. (2020). Settling Into Semantic Space: An Ambiguity-Focused Account of Word-Meaning Access. *Perspectives on Psychological Science*, 15(2):411–427. PMID: 31961780.

Rodd, J. M., Davis, M. H., and Johnsrude, I. S. (2005). The Neural Mechanisms of Speech Comprehension: fMRI studies of Semantic Ambiguity. *Cerebral Cortex*, 15(8):1261–1269.

Rodd, J. M., Johnsrude, I. S., and Davis, M. H. (2010). The role of domain-general frontal systems in language comprehension: Evidence from dual-task interference and semantic ambiguity. *Brain and language*, 115(3):182–188.

Rohde, H. and Kehler, A. (2014). Grammatical and information-structural influences on pronoun production. *Language, Cognition and Neuroscience*, 29(8):912–927.

Rosa, E. C. and Arnold, J. E. (2017). Predictability affects production: Thematic roles can affect reference form selection. *Journal of Memory and Language*, 94:43–60.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Ruder, S. (2019). *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway.

Saphra, N. and Lopez, A. (2019). Understanding Learning Dynamics Of Language Models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117.

Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Schütze, H. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shwartz, V. and Dagan, I. (2019). Still a Pain in the Neck: Evaluating Text Representations on Lexical Composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Small, S. L., Cottrell, G. W., and Tanenhaus, M. K. (2013). *Lexical Ambiguity Resolution: Perspective from Psycholinguistics, Neuropsychology and Artificial Intelligence*. Elsevier.

Smith, N. J. and Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Song, J. and Kaiser, E. (2020). Forward-looking Effects in Subject Pronoun Interpretation: What Comes Next Matters. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society (CogSci 2020)*.

Sorodoc, I.-T., Gulordava, K., and Boleda, G. (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.

Stevenson, R. J., Crawley, R. A., and Kleinman, D. (1994). Thematic roles, focus and the representation of events. *Language and cognitive processes*, 9(4):519–548.

Sturt, P., Pickering, M. J., and Crocker, M. W. (1999). Structural change and reanalysis difficulty in language comprehension. *Journal of Memory and Language*, 40(1):136–150.

Tanenhaus, M. K., Leiman, J. M., and Seidenberg, M. S. (1979). Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, 18(4):427–440.

Tenney, I., Das, D., and Pavlick, E. (2019a). BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *Proceedings of 7th International Conference on Learning Representations*.

Thater, S., Dinu, G., and Pinkal, M. (2009). Ranking Paraphrases in Context. In *Proceedings of the 2009 Workshop on Applied Textual Inference (TextInfer)*, pages 44–47, Suntec, Singapore. Association for Computational Linguistics.

Thater, S., Fürstenau, H., and Pinkal, M. (2011). Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Tily, H. and Piantadosi, S. (2009). Refer efficiently: Use less infofrrmative expressions for more predictable meanings. In *Proceedings of the Workshop on the Production of Referring Expressions: Bridging the Gap between Computational and Empirical Approaches to Reference (PRE-CogSci 2009)*.

Trott, S. and Bergen, B. (2021). RAW-C: Relatedness of Ambiguous Words–in Context (A New Lexical Resource for English). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 7077–7087. Association for Computational Linguistics.

Trueswell, J. C. and Tanenhaus, M. K. (1994). Toward a lexicalist framework of constraint-based syntactic ambiguity resolution. *Perspectives on Sentence Processing*.

Upadhye, S., Bergen, L., and Kehler, A. (2020). Predicting Reference: What do Language Models Learn about Discourse Models? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 977–982, Online. Association for Computational Linguistics.

Van Schijndel, M. and Linzen, T. (2018). Modeling garden path effects without explicit hierarchical syntax. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society (CogSci 2018)*.

Van Schijndel, M. and Linzen, T. (2021). Single-Stage Prediction Models Do Not Explain the Magnitude of Syntactic Disambiguation Difficulty. *Cognitive Science*, 45(6):e12988.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Vitello, S. and Rodd, J. M. (2015). Resolving semantic ambiguities in sentences: Cognitive processes and brain mechanisms. *Language and Linguistics Compass*, 9(10):391–405.

Voita, E., Sennrich, R., and Titov, I. (2019). The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. 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 4387–4397.

Wasow, T. (2015). Ambiguity avoidance is overrated. In *Ambiguity*, pages 29–48. de Gruyter.

Wasow, T., Perfors, A., and Beaver, D. (2005). The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe*, pages 265–282.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Wei, J., Pham, K., O'Connor, B., and Dillon, B. W. (2018). Evaluating Grammaticality in Seq2seq Models with a Broad Coverage HPSG Grammar: A Case Study on Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 298–305, Brussels, Belgium. Association for Computational Linguistics.

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., and Houston, A. (2013). Ontonotes release 5.0 ldc2013t19. Web Download.

Westera, M. and Boleda, G. (2019). Don't Blame Distributional Semantics if it can't do Entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 120–133, Gothenburg, Sweden. Association for Computational Linguistics.

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. *arXiv preprint arXiv:1909.10430*.

Wilcox, E., Levy, R., Morita, T., and Futrell, R. (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.

Wilson, D. and Carston, R. (2007). A unitary approach to Lexical Pragmatics: Relevance, Inference and Ad hoc concepts. *Pragmatics*, pages 230–259.

Wilson, D. and Sperber, D. (2006). Relevance Theory. *The Handbook of Pragmatics*, pages 606–632.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M.

(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.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J. R., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G. S., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yu, L. and Ettinger, A. (2020). Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4896–4907. Association for Computational Linguistics.

Zarcone, A., van Schijndel, M., Vogels, J., and Demberg, V. (2016). Salience and Attention in Surprisal-Based Accounts of Language Processing. *Frontiers in Psychology*, 7:844.

Zhou, W., Ge, T., Xu, K., Wei, F., and Zhou, M. (2019). BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Zipf, G. K. (1949). *Human behavior and the principle of least effort: An introduction to human ecology*. Addison-Wesley Press.

Zwaan, R. A. and Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, 123(2):162.