Madly Ambiguous:
A Game for Learning about Structural Ambiguity and Why It's Hard for Computers

Ajda Gokcen
Department of Linguistics
University of Washington
Seattle, WA 98195
ajdag@uw.edu

Ethan Hill
IBM Watson Health
Cleveland, OH 44114
ehill@us.ibm.com

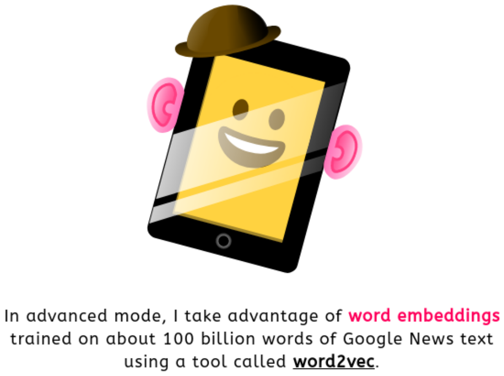
Michael White
Department of Linguistics
The Ohio State University
Columbus, OH 43210
mwhite@ling.osu.edu

Abstract

Madly Ambiguous is an open source, online game aimed at teaching audiences of all ages about structural ambiguity and why it's hard for computers. After a brief introduction to structural ambiguity, users are challenged to complete a sentence in a way that tricks the computer into guessing an incorrect interpretation. Behind the scenes are two different NLP-based methods for classifying the user's input, one representative of classic rule-based approaches to disambiguation and the other representative of recent neural network approaches. Qualitative feedback from the system's use in online, classroom, and science museum settings indicates that it is engaging and successful in conveying the intended take home messages.

1 Introduction

Madly Ambiguous is an open source,[1] in-browser online game[2] aimed at teaching audiences of all ages about structural ambiguity and some of the difficulties it poses for natural language processing. Users are introduced to the spunky Mr. Computer Head (Figure 1), a character who gives them an introduction to structural ambiguity and then challenges them to complete a sentence with a prepositional phrase attachment ambiguity in a way that he will misinterpret (Figures 4–5). After playing a round of the game, users may read more about how Mr. Computer Head and systems like him are trained to deal with tasks of ambiguity. Bringing Madly Ambiguous to fruition required an integration of NLP capabilities, cross-platform compatibility, and accessible pedagogical explanation of some fairly complex linguistic and computational concepts, the last of which proved to be the biggest challenge.



In advanced mode, I take advantage of **word embeddings** trained on about 100 billion words of Google News text using a tool called **word2vec**.

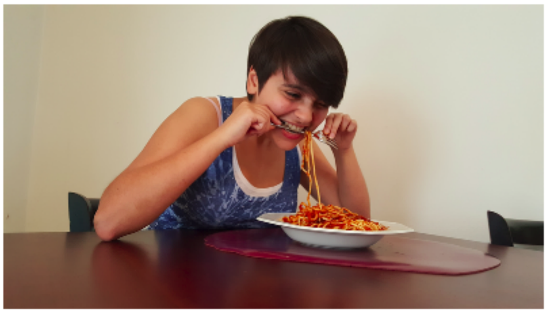
Figure 1: Mr. Computer Head, who acts as the opponent in the game, also narrates the introduction and explanation.

Madly Ambiguous has been developed as an outreach component of a project whose aim is to develop methods for avoiding ambiguity in natural language generation and for using disambiguating paraphrases to crowd source interpretations of structurally ambiguous sentences (Duan and White, 2014; Duan et al., 2016; White et al., 2017). The game was initially intended solely as an iPad demo outside of Ohio State's Language Sciences Research Lab, or "Language Pod," a fully functional research lab embedded within the Columbus Center of Science and Industry (COSI), one of the premier science centers in the country (Wagner et al., 2015). The COSI research pods are glass-enclosed research spaces where museum visitors can observe actual scientific research as it is occurring, creating excitement in children about science and encouraging scientific careers. Outside the pod, Ohio State graduate and undergraduate students (the "explainers") provide educational explanations to both adult and child COSI visitors about the work being conducted within the pod as well as language science in general. The explain-

[1] https://github.com/ajdagokcen/madlyambiguous-repo

[2] http://madlyambiguous.osu.edu

Proceedings of NAACL-HLT 2018: Demonstrations, pages 51–55
New Orleans, Louisiana, June 2 - 4, 2018. ©2018 Association for Computational Linguistics



Ouch — that's a little TOO al dente!

Figure 2: An illustration of why interpreting the sentence *Jane ate spaghetti with a fork* as the fork being part of the dish (instead of a utensil) is ridiculous and easy for a human to dismiss (albeit still a potential source of confusion for a computer).

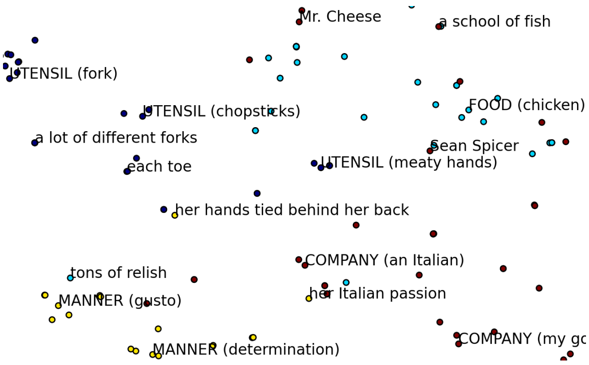


Figure 3: A zoomed-in view of a t-SNE plot showing some of the clusters of similar input phrases to the word2vec model. Phrases like "an Italian" and "her Italian passion" have different interpretations (COMPANY and MANNER, respectively), but are very close to one another on this plot, showing that even the more advanced method has its difficulties.

ers receive extensive training on how to talk about science to a general audience via courses offered at OSU and also from the COSI educational team.

The Language Pod organizers were enthusiastic about the development of Madly Ambiguous since they were aware of no general audience demos that dealt with syntax-related linguistic phenomena. After gathering feedback on the initial iPad version of Madly Ambiguous at COSI, it was completely redesigned as an in-browser demo that can be used on iPad, Android, and desktop browsers, both for informal science learning and undergraduate classroom use, as well as a stand-alone demo on the web. Qualitative feedback on the revamped Madly Ambiguous suggests that it is educational and engaging for all ages.

2 Design

2.1 Interface

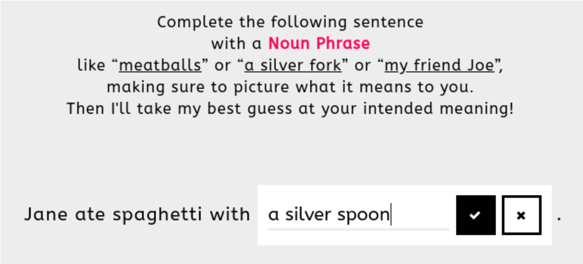
Madly Ambiguous's interface is implemented using Node.js[3] as a single dynamic web page. It includes three primary sections: the introduction, the game, and the explanation of how it works. The introduction discusses the more general principles of structural ambiguity as well as the particular rules of the game, including interactive elements and humorous examples to make the instructions more interesting (Figure 2). The explanation of how it works can be read once the user has gone through at least one round of the game; it gives the basics of the two different methods the system uses for classifying the input, as described further in the next subsection.

The game itself has two phases of user interaction. First, users fill in the blank in the sentence, "Jane ate spaghetti with __ ." (See Figure 4.) The system gives a waiting screen depicting a contemplative Mr. Computer Head as it processes and classifies their input, and then displays the guess for users to confirm or deny based on their intended interpretation, as shown in Figure 5. Four different interpretations are possible, with one additional selection if the user feels none of the four capture the meaning. Once the user selects an answer, s/he is given the option to play again, possibly switching between basic and advanced mode.

As we discovered during trials of the initial version of the system, the main challenge of the interface was in presenting the different possible interpretations to users in a way that those with no prior understanding of linguistics could quickly grasp. In the current version, this is accomplished by presenting each option not just with a paraphrase of the sentence that captures the same meaning in a less ambiguous way, but also with a picture depicting the interpretation. Note that the pictures that accompany each meaning are based on the sentences in the introduction as opposed to the user's input, so even if the user enters a utensil such as *a silver spoon*, the picture for the UTENSIL interpretation always shows a fork.

Given the importance of illustrative pictures in making the demo accessible, along with the difficulty of staging such pictures, the current version includes only the sentence for which we have corresponding photos for each interpretation.

[3] https://nodejs.org



Complete the following sentence
with a Noun Phrase
like "meatballs" or "a silver fork" or "my friend Joe",
making sure to picture what it means to you.
Then I'll take my best guess at your intended meaning!

Jane ate spaghetti with a silver spoon ✓ × .

Figure 4: The main screen of the game, where users are asked to complete the ambiguous sentence in a way that the system will misinterpret.

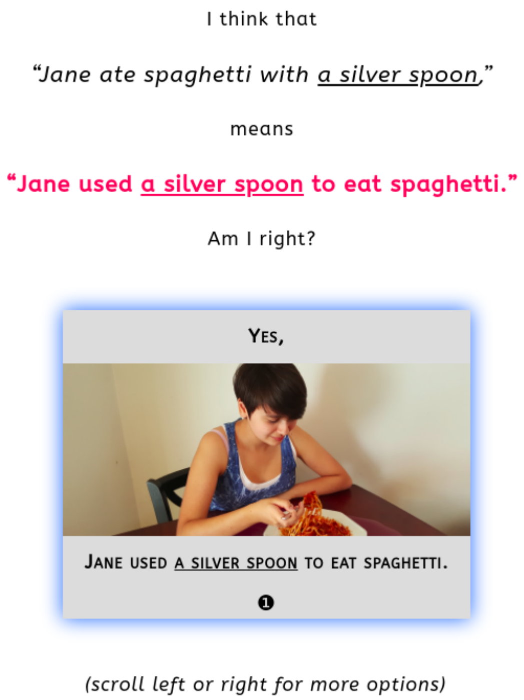
2.2 The NLP

Behind the scenes, the system classifies the user's completion of the sentence "Jane ate spaghetti with _ " as having one of the following four semantic roles, as represented by keywords and paraphrases:

- UTENSIL: *Jane used _ to eat spaghetti.*
- PART: *Jane had spaghetti and _ .*
- MANNER: *Jane exhibited _ while eating spaghetti.*
- COMPANY: *Jane ate spaghetti in the presence of _ .*

There are two different methods of analysis that can be employed. *Basic mode* represents a classic rule-based approach to NLP, utilizing part-of-speech tagging, lemmatization, and WordNet (Miller, 1995) to arrive at an answer. This requires some heuristics based on the part-of-speech tags and lemmas in order to decide what the "most important" word of the input phrase is for cases like *Jane ate spaghetti with a bowl full of meatballs*. The most important word (or multiword phrase) is then looked up in WordNet and its hypernyms are used to choose the category, much as with how selectional restrictions have been traditionally used (e.g. Allen et al., 2001).

Advanced mode uses methods closer to the current state-of-the-art for modern NLP, namely word embeddings (Mikolov et al., 2013). The gensim implementation of word2vec (Řehůřek and Sojka, 2010) is used with vectors that have been pre-trained on the Google News corpus. A training set of phrases and interpretation labels is used to create clusters for each of the four interpretations. Inputs are then classified based on the nearest neighbor in the model to the average of all of the word vectors in the input phrase, not unlike in recent memory-based approaches to one-shot learning (Vinyals et al., 2016).



I think that

"Jane ate spaghetti with a silver spoon."

means

"Jane used a silver spoon to eat spaghetti."

Am I right?

YES,

JANE USED A SILVER SPOON TO EAT SPAGHETTI.

❶

(scroll left or right for more options)

Figure 5: Once users complete the sentence, they're shown the system's guess for the sentence's meaning and can confirm or deny its veracity.

The explanation of how it works additionally covers common sources of interpretation errors. In basic mode, infrequent word senses listed in WordNet can cause confusion; for example, *trump* is listed as an archaic form of *trumpet*, leading Mr. Computer Head to conjecture that *President Trump* is a utensil. In advanced mode, the blending of unrelated senses in word embeddings can cause trouble; for example, as shown in the visualization of the clusters in Figure 3, the food and manner senses represented in the embedding for *relish* can lead to mistakes, as *tons of relish* is closer to one of the MANNER cluster centroids than the intended FOOD clusters.

3 Educational Objectives and Feedback

For informal science learning, like at COSI, the presentation of Madly Ambiguous can and should be tailored to different audiences. For all ages, the critical take home message is that sentences can have more than one meaning (even when the meaning of the words remains constant), and that while people are adept at using the context to determine what's intended, this can be very hard for computers.[4] Depending on the audience, the explainers might also skip the intro and jump right into the game with the pitch, *Hey, do you want to try to trick a computer?*

To separate the notion of intended meaning from the form of the sentence, users of the demo are encouraged to visualize the meaning they have in their head before clicking to see how Mr. Computer Head interprets their sentence completion. With more advanced audiences, the explainers will discuss how linguists use technical tools (like dependency trees) to analyze structural ambiguities and go over how the basic and advanced mode work. Finally, by discussing the kinds of errors the system makes, the explainers can broach the topic of why computers remain so much worse at ambiguity resolution than people. Classroom use can be similar, but with more background knowledge, students can be challenged to come up with ways to improve upon the system's current strategies.

Since the demo went live in Summer 2017, Mr. Computer Head's accuracy against user judgments is currently 64% for basic mode and 70% for advanced mode, well above the majority baseline of 29% despite most users trying hard to fool him.[5] Qualitatively, a high level of engagement with the demo can be observed by examining the lengths to which users go to win, cleverly coming up examples like *a cucumber dressed as a person* as COMPANY rather than FOOD, *pins and needles* as MANNER rather than UTENSIL, and *very British reserve* as MANNER rather than COMPANY, all of which fool Mr. Computer Head in one mode or the other.

Madly Ambiguous received more widespread community feedback after popular linguistics blog *All Things Linguistic* made a post about it, describing it as "a nice intro to automatic sentence processing" (McCulloch, 2017). From there the link was shared across Twitter, Facebook, and beyond. Translation platform Smartcat reached out to learn more about computational linguistics in a webcast interview (Banffy, 2017; Academy, 2017), while other computational linguistics pages like UW-CLMS discussed it on Facebook (CLMS, 2017).

Teachers of courses related to language and computers have also made posts about using Madly Ambiguous in the classroom, making comments such as, "I actually cannot believe I showed word2vec visualizations in a 100 level course – some people were at least nodding, and they are not all from [a] CS background. Absolutely loved using it as a pedagogical tool, and the students also seemed to have understood better" (Vajjala, 2017).

4 Summary and Future Work

In this paper we have introduced Madly Ambiguous, a game aimed at teaching audiences of all ages about structural ambiguity and demonstrating why it's hard for computers—an important lesson that serves to demystify natural language processing at a time when AI in general is arguably overhyped, risking societal overreactions to the technology. Although Madly Ambiguous is complete and publicly available as-is, there are still more directions it could be taken in, as well as improvements to be made. Since the system saves the data from each round played, there are, as of February 2018, over 13,000 user inputs and judgments collected, which could be used as dynamic feedback for training future versions or possibly as data for other studies of structural ambiguity.

The game could be extended to include other sentences and types of structural ambiguity, such as with coordination (e.g., *The old dogs and cats went to the vet*, where *old* may modify *dogs and cats* or *dogs* alone). This may call for additional illustrative pictures, however. Other expansions might incorporate different successful vector-based methods into the word2vec mode to make it even more sophisticated. Compositional character models, as in Ling et al. (2015), could allow the system to meaningfully model even out-of-vocabulary words; syntactically/semantically compositional models as in Socher et al. (2012) could yield a single vector for multi-word phrases that composes the representations for each word rather than averaging them, potentially providing more separation between clusters. Another direction would be to dynamically generate explanations. It is an open source project, so anyone could contribute to the code!

Acknowledgments

We thank David King, Matt Metzger, and Kaleb White for their contributions to the interface and advanced mode functionality. The interactive demo materials were contributed by Laura Wagner and Victoria Sevich. Special thanks also to Kathryn Campbell-Kibler and Christy Doran for helpful suggestions, and to Yasemin Gokcen and Jessica Findsen for modeling with spaghetti as Jane and Mary. Madly Ambiguous was funded in part through NSF Grant 1319318.

[4] Indeed, PP-attachment ambiguities have remained a primary source of parser errors (Kummerfeld et al., 2012).

[5] To our surprise, the youngest users of Madly Ambiguous often want to *help* Mr. Computer Head get the right answer!

References

Smartcat Academy. 2017. Computational linguistics. https://www.crowdcast.io/e/computational-linguistics.

James F Allen, Donna K Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Toward conversational human-computer interaction. *AI magazine*, 22(4):27.

Octávio Banffy. 2017. Madly ambiguous linguistic game. https://community.smartcat.ai/topic/792-madly-ambiguous-linguistic-game/.

UW CLMS. 2017. Uw professional master's in computational linguistics: A fun game from ohio state. https://www.facebook.com/uwclma/posts/10155182653273246.

Manjuan Duan, Ethan Hill, and Michael White. 2016. Generating disambiguating paraphrases for structurally ambiguous sentences. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 160–170, Berlin, Germany. Association for Computational Linguistics.

Manjuan Duan and Michael White. 2014. That's Not What I Meant! Using Parsers to Avoid Structural Ambiguities in Generated Text. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 413–423, Baltimore, Maryland. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, South Korea. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530.

Gretchen McCulloch. 2017. Madly ambiguous is a fill-in-the-blank game that teaches you about ambiguity while you try to trick a computer. http://allthingslinguistic.com/post/165950061882/madly-ambiguous-is-a-fill-in-the-blank-game-that.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics.

Sowmya Vajjala. 2017. Teaching Notes — Teaching about 'What is NLP?'. https://nishkalavallabhi.github.io/LandC5/.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. *CoRR*, abs/1606.04080.

Laura Wagner, Shari R. Speer, Leslie C. Moore, Elizabeth A. McCullough, Kiwako Ito, Cynthia G. Clopper, and Kathryn Campbell-Kibler. 2015. Linguistics in a science museum: Integrating research, teaching, and outreach at the language sciences research lab. *Language and Linguistics Compass*, 9(10):420–431.

Michael White, Manjuan Duan, and David L. King. 2017. A simple method for clarifying sentences with coordination ambiguities. In *Proceedings of the 1st Workshop on Explainable Computational Intelligence (XCI 2017)*. Association for Computational Linguistics.